

INFO PRICE \$ \_\_\_\_\_

POST PRICE(S) \$ \_\_\_\_\_

E 20-177  
9 MARCH 1966  
E 20-177  
9 MARCH 1966

Hard copy (HC) 4.00

Microfiche (MF) .75

Interim Technical Report

H 653 J IV 65

# SIMULTANEOUS SOLUTION OF NON-LINEAR ALGEBRAIC EQUATIONS

Research & Analysis Section Tech. Memo No. 163

*Prepared for:*

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
GEORGE C. MARSHALL SPACE FLIGHT CENTER  
Aero - Astrodynamics Laboratory**

**PREPARED UNDER CONTRACT No. NAS8-20183**

FACILITY FORM 402	<u>N66 24937</u> (ACCESSION NUMBER)	_____ (THRU)
	<u>111</u> (PAGES)	_____ (CODE)
	<u>CR-74771</u> (NASA CR OR TMX OR AD NUMBER)	<u>19</u> (CATEGORY)
	_____ (OTHER NUMBER)	_____

**NORTHROP SPACE LABORATORIES**

6025 TECHNOLOGY DRIVE, HUNTSVILLE, ALABAMA 35805  
TELEPHONE 837-0550  
P. O. BOX 1484

INTERIM TECHNICAL REPORT

SIMULTANEOUS SOLUTION OF

NON-LINEAR ALGEBRAIC EQUATIONS

Research & Analysis Section Tech. Memo No. 163

Prepared for:

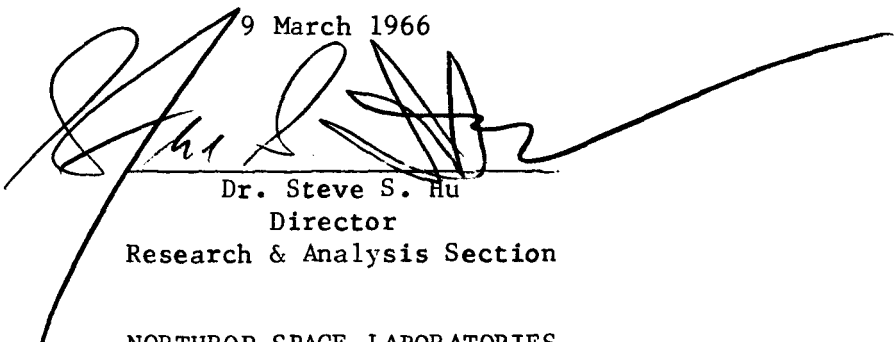
National Aeronautics and Space Administration  
George C. Marshall Space Flight Center  
Aero-Astroynamics Laboratory

Under Contract No. NAS8-20183

by

Frank B. Tatom  
Virginia C. Lillard  
Lalit M. Bhalla

9 March 1966



Dr. Steve S. Hu  
Director  
Research & Analysis Section

NORTHROP SPACE LABORATORIES  
HUNTSVILLE DEPARTMENT  
HUNTSVILLE, ALABAMA

INTERIM TECHNICAL REPORT  
Contract NAS8-20183

SIMULTANEOUS SOLUTION OF  
NON-LINEAR ALGEBRAIC EQUATIONS

By

Frank B. Tatom  
Virginia C. Lillard  
Lalit M. Bhalla

Northrop Space Laboratories  
Huntsville, Alabama

ABSTRACT

24937

This report presents the results of a research effort concerned with the development of a general digital computer program capable of solving sets of nonlinear algebraic equations which are characteristic of filter circuit analysis. The program utilizes a combination of the Newton-Raphson method and the Freudenstein-Roth technique. Results of computer runs involving six equations and unknowns and thirteen equations and unknowns are presented.

The report indicates that the computer program developed is capable of solving sets of nonlinear algebraic equations, and the ease with which the solution is achieved is directly related to the accuracy of the initial estimates of the roots to the equations.

*Author*

TABLE OF CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
1.0	INTRODUCTION. . . . .	1
2.0	TECHNICAL DISCUSSION. . . . .	2
	2.1 Statement of the Problem . . . . .	2
	2.2 Physical Considerations. . . . .	5
	2.3 Application of Numerical Techniques. . . . .	9
	2.4 Selection of Initial Estimates . . . . .	20
	2.5 Digital Computer Considerations. . . . .	22
3.0	DIGITAL COMPUTER PROGRAM. . . . .	24
	3.1 Basic Features . . . . .	24
	3.2 Inputs and Options . . . . .	30
	3.3 Outputs. . . . .	32
4.0	DISCUSSION OF RESULTS . . . . .	33
	4.1 Relative Performance of Numerical Methods. . . . .	33
	4.2 Application to Actual Problems . . . . .	34
5.0	CONCLUSIONS AND RECOMMENDATIONS . . . . .	39
	APPENDIX A SOURCE PROGRAM. . . . .	40
	APPENDIX B SUBROUTINES . . . . .	65
	APPENDIX C INPUTS AND OUTPUTS. . . . .	69
	APPENDIX D FILTER CIRCUIT WITH SIX UNKNOWNNS. . . . .	80
	APPENDIX E FILTER CIRCUIT WITH THIRTEEN UNKNOWNNS . . . . .	82
	REFERENCES CITED. . . . .	85

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
2-1	Range of Values for Filter Circuit Components . . . . .	7
4-1	Results of Computer Solution of Six Equations . . . . .	35
4-2	Results of Computer Solution of Thirteen Equations. . . . .	36

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
2-1	Typical Electronic Filter Circuit. . . . .	3
3-1a	Program for Solution of Nonlinear Algebraic Equations. . . .	25
3-1b	Subroutine ESTIM . . . . .	26
3-1c	Subroutine SIMEQ . . . . .	27
3-1d	Subroutine FCON. . . . .	28

LIST OF SYMBOLS

A. English Letters

<u>Symbol</u>	<u>Definition</u>
$A_j$	the coefficient of the first term of the $j^{\text{th}}$ equation which is systematically reduced to unity
$A_j^{(m)}$	the $m^{\text{th}}$ value of the coefficient of the $j^{\text{th}}$ equation
$a_{1n}$	the superscript for $Y_n$ corresponding to the minimum value of $\sum_{j=1}^P  \psi_j^{(m,n)} - F_j $ for the first variation of $Y_n^{(m)}$
$a_{2n}$	the superscript for $Y_n$ corresponding to the minimum value of $\sum_{j=1}^P  \psi_j^{(m,n)} - F_j $ for the second variation of $Y_n^{(m)}$
$C_m$	the $m^{\text{th}}$ capacitance expressed in farads
$D_q$	$q^{\text{th}}$ coefficient in series in denominator of transfer function
$d_j$	the degree of the $j^{\text{th}}$ equation
$F_j$	the constant in the $j^{\text{th}}$ equation
$F_j^{(m)}$	the $m^{\text{th}}$ value of the constant in the $j^{\text{th}}$ equation
$G$	number of terms in the numerator of the transfer function
$H$	number of terms in the denominator of the transfer function
$K_m$	a constant relating the $m^{\text{th}}$ natural resistance with the $m^{\text{th}}$ inductance
$L_m$	the $m^{\text{th}}$ inductance expressed in henries

<u>Symbol</u>	<u>Definition</u>
$N_q$	$q^{\text{th}}$ coefficient in series in numerator of transfer function
$p$	the number of equations or unknowns
$Q_j$	the number of terms in the $j^{\text{th}}$ equation
$Q_{j(\text{max})}$	the number of terms in the longest equation
$Q_{\text{limit}}$	the number of times the columns of terms in the set of equations are rearranged
$R_m$	the $m^{\text{th}}$ resistance expressed in ohms
$R_m^{(b)}$	natural resistance for the $m^{\text{th}}$ inductance ( $m=1,2,\dots,v$ ) expressed in ohms
$R_m^{(s)}$	surplus resistance in the $m^{\text{th}}$ resistance $R_m$ ( $m=1,2,\dots,v$ ) expressed in ohms
$r(j,i,k)$	the subscript for the $k^{\text{th}}$ factor in the $i^{\text{th}}$ term of the $j^{\text{th}}$ equation
$s$	Laplace transform variable
$T$	transfer function
$t_{ji}$	the $i^{\text{th}}$ term in the $j^{\text{th}}$ equation
$U_{\text{limit}}$	maximum number of steps to be used in the Newton-Raphson method
$u$	number of resistances in the circuit
$V$	the selected number of iterative steps in the Freudenstein-Roth technique



<u>Symbol</u>	<u>Definition</u>
$V_{\text{limit}}$	maximum number of steps in the Freudenstein-Roth technique
$v$	number of inductances in the circuit
$W$	number of steps selected for use with the process of selection of initial estimates
$w$	number of capacitances in the circuit
$X_n$	the $n^{\text{th}}$ unknown defined by Eq. (2-14)
$X_n^{(m)}$	$m^{\text{th}}$ estimate of $X_n$
$\vec{X}^{(m)}$	the position vector representing the point $(X_1^{(m)}, X_2^{(m)}, \dots, X_p^{(m)})$
$Y_m^{(b)}$	unknown corresponding to $R_m^{(b)}$ ( $m=1,2,\dots,v$ )
$Y_m^{(s)}$	unknown corresponding to $R_m^{(s)}$ ( $m=1,2,\dots,v$ )
$Y_n$	circuit elements (resistances, inductances, and reciprocals of capacitances) or unknown magnitude
$\bar{Y}_n$	$\left[ Y_{n(\text{low})} \cdot Y_{n(\text{up})} \right]^{\frac{1}{2}}$
$Y_n^{(g)}$	final value of $Y_n$ resulting from the process of selection of initial estimates
$Y_n^{(m)}$	$m^{\text{th}}$ value of $Y_n$ obtained during the process of selecting initial estimates or other iterative process
$Y_{n(\text{low})}$	lower limit of the range of interest for $Y_n$
$Y_{n(\text{up})}$	upper limit of the range of interest for $Y_n$

B. Greek Letters

<u>Symbol</u>	<u>Definition</u>
$\Delta a$	an integer usually taken as unity which represents convergence criteria in the process of selection of initial estimates
$\Delta X_n^{(m)}$	$(X_n^{(m+1)} - X_n^{(m)})$
$\epsilon_j^{(m)}$	the $m^{\text{th}}$ value of the $j^{\text{th}}$ residual
$\lambda$	a variable whose magnitude must satisfy Eq. (2-28)
$\tau$	the function defined by Eq. (2-22)
$\nabla \tau$	the gradient of the function $\tau$
$\phi_j^{(m)}$	$\phi_j(X_1^{(m)}, X_2^{(m)} \dots X_p^{(m)})$
$\psi_j$	the $j^{\text{th}}$ function of the form of Eq. (2-3)
$\psi_j^{(m,n)}$	the value of $\psi_j$ obtained in the $m^{\text{th}}$ step of the variation of $Y_n^{(m)}$ in the process of selection of initial estimates

SUMMARY

A research effort has been carried out to develop a general digital computer program which is capable of simultaneously solving sets of nonlinear algebraic equations which arise in problems involving filter circuit analysis. A number of different numerical techniques were considered. A combination of the Newton-Raphson method and the Freudenstein-Roth technique was found to be the most promising approach. To increase the latitude in the selection of initial estimates of the roots to the equations, a special iterative selection process was developed to be used in conjunction with the combined Newton-Raphson Freudenstein-Roth technique.

The numerical methods were incorporated into a digital computer program which was tested on sets of equations consisting of six equations and thirteen equations. The program successfully solved each set for two sets of roots which appeared to "cluster" together to some extent. For the larger set of equations, some difficulty was encountered in achieving convergence when the initial estimates were more than one order of magnitude from the actual roots.

The conclusion was reached that the program, with the combined numerical methods already mentioned, is suitable for solution of sets of nonlinear algebraic equations. Although the program is specifically designed to handle equations characteristic of filter circuit analysis, it was noted that only minor modifications would be necessary to permit application of the program to any set of nonlinear algebraic equations. The possibility of certain numerical refinements to the program appeared likely. Also, consideration of

**NORTHROP SPACE LABORATORIES**

---

the actual availability of electronic components corresponding to the roots obtained seemed worthy of study. The recommendation was made that the research effort be continued to investigate such items.

## 1.0 INTRODUCTION

In filter circuit analysis, problems arise which involve the simultaneous solution of sets of nonlinear algebraic equations. Solution of such sets by hand can be extremely difficult, and if a large number of equations are involved, hand calculations become impractical. The use of digital computers, coupled with appropriate numerical techniques, is a logical approach to such problems. In developing the necessary digital computer program, consideration must be given to the fact that there are many different filter circuits, and the set of equations corresponding to any one circuit will generally be different from all other sets. Thus the most desirable program is the most general, capable of solving a number of different sets of equations.

The Huntsville Department of Northrop Space Laboratories, under Contract NAS8-20183 for the Aero-Astroynamics Laboratory of Marshall Space Flight Center, has been engaged in the development of a general digital computer program capable of solving sets of nonlinear algebraic equations. Mr. Mario Rheinfurth of the Control Theory Branch, Dynamics and Flight Mechanics Division, has acted as the NASA Technical Representative for this research effort. Section 2.0 of this report provides a detailed description of the problem involved, applicable numerical schemes, and digital computer considerations. A discussion of the computer program is presented in Section 3.0. Section 4.0 describes the results obtained. Conclusions and recommendations are presented in Section 5.0.

## 2.0 TECHNICAL DISCUSSION

### 2.1 Statement of the Problem

Transfer functions associated with electronic filter circuits, such as that shown in Figure 2-1, have the general form

$$T = \frac{\sum_{q=1}^G N_{q-1} s^{q-1}}{\sum_{q=1}^H D_{q-1} s^{q-1}} \quad (2-1)$$

where

T = transfer function

G = number of terms in numerator

$N_q$  =  $q^{\text{th}}$  coefficient in series in numerator

s = Laplace transform variable

H = number of terms in denominator

$D_q$  =  $q^{\text{th}}$  coefficient in series in denominator.

Generally, the numerical values of the coefficients,  $N_q$  and  $D_q$ , are obtained by curve fitting. Based on circuit analysis of a given filter circuit, a set of equations can be derived. The derivation can be accomplished by means of a flow graph (ref. 1), which makes use of Kirchhoff's Law, or by use of topology (ref. 2). The resulting set of equations may be written as

$$\psi_j(Y_1, Y_2, \dots, Y_n \dots Y_p) = F_j \quad (j=1, 2, \dots, p) \quad (2-2)$$

where

$$p = G + H$$

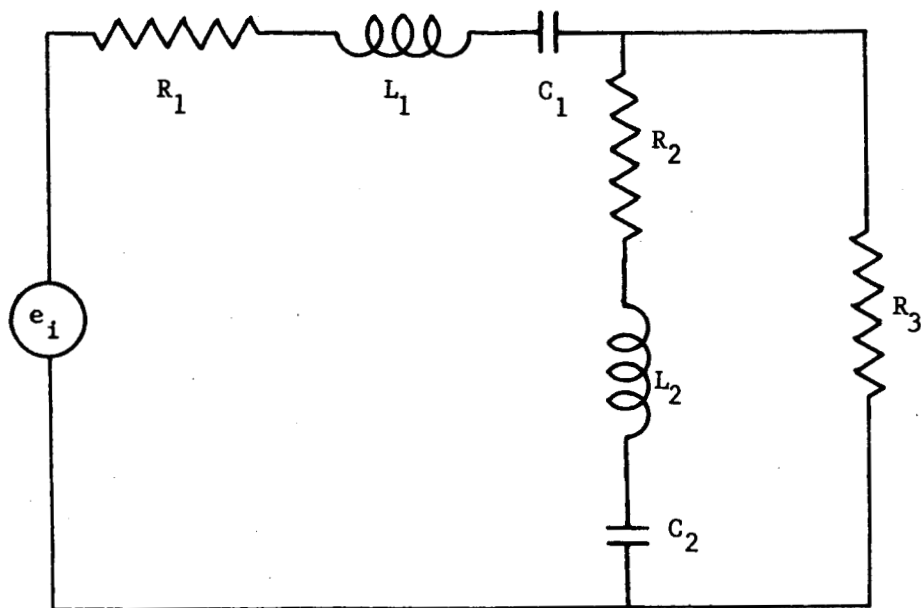


Figure 2-1. TYPICAL ELECTRONIC FILTER CIRCUIT

$$F_j = \begin{cases} N_{j-1} & (j=1,2,\dots,G) \\ D_{j-G-1} & (j=G+1\dots G+H) \end{cases}$$

$Y_n$  = circuit elements (resistances, inductances, and reciprocals of capacitances) of unknown magnitude.

Thus for each coefficient  $N_q$  or  $D_q$  there is an equation in which the coefficient appears as a constant,  $F_j$ . It should be noted that although the magnitudes of the  $Y_n$  are unknown, their identities as resistances, inductances, or reciprocals of capacitances are established by the filter circuit analysis. The reciprocal of capacitance is used because the resulting form of  $\psi_j$  is easier to work with.

In Eq. (2-2) the functions  $\psi_j$  consist of a sum of terms of the form

$$\psi_j = \sum_{i=1}^{Q_j} t_{ji} \quad (j=1,2,\dots,p) \quad (2-3)$$

where the term  $t_{ji}$  has the form

$$t_{ji} = \prod_{k=1}^{d_j} Y_{r(j,i,k)} \quad (j=1,2,\dots,p) \quad (i=1,2,\dots,Q_j) \quad (2-4)$$

with

$$r(j,i,1) \neq r(j,i,2) \neq \dots \neq r(j,i,k) \neq \dots \neq r(j,i,d_j).$$

Notice should be taken that for each unknown in each term the value of  $r(j,i,k)$  is known. Furthermore, for any value of  $j$ , all terms are of the same degree,  $d_j$ . All of the functions,  $\psi_j$ , however, are not necessarily of the same degree.



In order to establish an orderly relationship between  $Y_n$  and the resistances, capacitances, and conductances, it is convenient to use the following arrangement:

$$\begin{aligned} Y_n &= R_n \quad (n=1,2,\dots,u) \\ Y_n &= L_{n+1-u} \quad (n=u+1\dots u+v) \\ Y_n &= 1/C_{n+1-u-v} \quad (n=u+v+1,\dots,u+v+w) \end{aligned} \quad (2-5)$$

where

$R_n$  = the  $n^{\text{th}}$  resistance

$L_n$  = the  $n^{\text{th}}$  inductance

$C_n$  = the  $n^{\text{th}}$  capacitance

$u$  = number of resistances in the circuit

$v$  = number of inductances in the circuit

$w$  = number of capacitances in the circuit.

It is important to note that, based on Eq. (2-2),

$$p = u + v + w \quad (2-6)$$

The basic objective is to determine the values of the circuit components,  $Y_n$ , which, subject to certain physical constraints, satisfy Eq. (2-2) for given values of  $F_j$ . Thus, the problem involves the simultaneous solution of sets of nonlinear equations with "p" equations and "p" unknowns.

## 2.2 Physical Considerations

Because the circuit elements are real positive numbers, the desired roots must likewise fall in this category. For practical purposes there exist minimum

and maximum values for resistances, inductances, and capacitances as indicated in Table 2-1. Thus, the roots to be obtained must fall in the appropriate range according to the identity of each unknown as a resistance, inductance, or reciprocal of capacitance.

Because each inductance in a circuit also has a "built-in" or natural resistance associated with it in series, consideration must be given to the functional relationship between each inductance and its natural resistance. In formulating Eq. (2-2), these natural resistances are treated as portions of unknown resistances, but actually they are each dependent on a particular inductance. Thus, in the circuit there exists  $v$  resistances which each contain a natural resistance. For ease in relating these resistances to the appropriate inductances it is convenient when numbering the circuit components to use the same numerical subscript for an inductance and the corresponding resistance. Thus  $R_1$  contains the natural resistance for  $L_1$ ,  $R_2$  the natural resistance for  $L_2$ , etc. In general, based on the relationships provided in Eqs. (2-5), the natural resistance for  $L_m$  where

$$L_m = Y_{u+m} \quad (m=1,2,\dots,v) \quad (2-7)$$

would be found in  $R_m$  where

$$R_m = Y_m \quad (m=1,2,\dots,v) \quad (2-8)$$

With the numbering arrangement outlined, all resistances with subscripts equal to or less than  $v$  are composed of two parts. One part is the natural resistance,  $R_m^{(b)}$  for an inductance, and the second part is a "surplus" resistance,  $R_m^{(s)}$ .

Table 2-1

Range of Values for Filter Circuit Components

<u>Component</u>	<u>Minimum</u>	<u>Maximum</u>
Resistances (ohms)	$2.4 \times 10^{-1}$	$2.2 \times 10^7$
Inductances (henries)	$5.0 \times 10^{-5}$	$3.5 \times 10^2$
Capacitances (farads)	$1.0 \times 10^{-11}$	$1.5 \times 10^{-1}$

Thus

$$R_m = R_m^{(b)} + R_m^{(s)} \quad (m=1,2,\dots,v) \quad (2-9)$$

or

$$Y_m = Y_m^{(b)} + Y_m^{(s)} \quad (m=1,2,\dots,v) \quad (2-10)$$

The functional relationship between an inductance and its natural resistance is dependent on the electrical characteristics and physical dimensions of the wire which makes up the inductance. For practical purposes, however, a linear relationship between inductance and natural resistance appears satisfactory.

Thus

$$R_m^{(b)} = K_m L_m \quad (m=1,2,\dots,v) \quad (2-11)$$

or

$$Y_m^{(b)} = K_m Y_{u+m} \quad (m=1,2,\dots,v) \quad (2-12)$$

where  $K_m$  = a constant (normally taken as unity).

Thus, by substitution,

$$Y_m = Y_m^{(s)} + K_m Y_{u+m} \quad (m=1,2,\dots,v) \quad (2-13)$$

From Eq. (2-13) it can be seen that for  $m=1,2,\dots,v$ ,  $Y_m^{(s)}$  are the true independent variables instead of  $Y_m$ . To avoid unnecessary use of superscripts, while at the same time positively identifying the true independent unknowns, a change of variable is convenient. Thus by definition,

$$X_n = \begin{cases} Y_n^{(s)} & (n=1,2\dots v) \\ Y_n & (n=v+1,v+2\dots p) \end{cases} \quad (2-14)$$

All previously mentioned physical constraints for  $Y_n$  apply also to  $X_n$ . In terms of the new variables,  $X_n$ , Eqs. (2-2) may be written

$$\phi_j(X_1, X_2, \dots, X_n, \dots, X_p) = F_j \quad (j=1,2\dots p) \quad (2-15)$$

An examination of Eqs. (2-15) reveals that while the form of functions has changed from  $\psi_j$  to  $\phi_j$  the problem remains essentially the same.

In addition to the relationship between inductances and resistances, there also exists the question of incrementation. The actual roots to Eqs. (2-2) may be of such nonintegral values that some or all of them cannot be exactly matched by circuit components which are normally fabricated. Thus the components of an actual filter circuit may only approximate the theoretical roots necessary to satisfy the appropriate equations for a given transfer function.

### 2.3 Application of Numerical Techniques

Several techniques have been widely used for some time for simultaneous solution of nonlinear algebraic equations. These include the Newton-Raphson method, the Method of Steepest Descent, and relaxation. More recently, several new approaches have been developed. One of the most promising is the Freudenstein-Roth technique (ref. 3).

In the research effort covered by this report, the relaxation method as described in reference 4 was found to be relatively incompatible with digital computer application. As a result, attention was primarily directed toward application of the Newton-Raphson method, the Method of Steepest Descent, and the Freudenstein-Roth technique.

### 2.3.1 The Newton-Raphson Method

Probably the most widely used method for solving simultaneous nonlinear algebraic equations, as well as transcendental equations, is the Newton-Raphson method. The method is described in various numerical analysis texts (refs. 5-8) and thus only a brief description need be given here.

The Newton-Raphson method is a successive-approximation procedure. Based on an initial estimate of the unknowns,  $x_n^{(0)}$ , the values of the  $\phi_j^{(0)}$  are calculated and compared with the values  $F_j$ . The difference is referred to as the residual,  $\epsilon_j^{(0)}$ . Thus

$$\epsilon_j^{(0)} = \phi_j^{(0)} - F_j \quad (2-16)$$

where

$$\phi_j^{(0)} = \phi_j(x_1^{(0)}, x_2^{(0)}, \dots, x_p^{(0)}),$$

or in general

$$\epsilon_j^{(m)} = \phi_j^{(m)} - F_j \quad (2-17)$$

where

$$\phi_j^{(m)} = \phi_j(X_1^{(m)}, X_2^{(m)}, \dots, X_p^{(m)}),$$

and

$$X_n^{(m)} = m^{\text{th}} \text{ estimate of } X_n.$$

Obviously, when the residuals are all simultaneously zero, the solution has been achieved. A first-order series expansion for each of the functions  $\phi_j$  about the initial estimate is used to approximate the functions. Thus

$$\phi_j \approx \phi_j^{(0)} + \sum_{n=1}^p \frac{\partial \phi_j^{(0)}}{\partial X_n} [X_n^{(1)} - X_n^{(0)}] \quad (2-18)$$

or by Eq. (2-15)

$$F_j \approx \phi_j^{(0)} + \sum_{n=1}^p \frac{\partial \phi_j^{(0)}}{\partial X_n} [X_n^{(1)} - X_n^{(0)}] \quad (2-19)$$

Based on the definition of the residual as given by Eq. (2-16),

$$\epsilon_j^{(0)} = - \sum_{n=1}^p \frac{\partial \phi_j^{(0)}}{\partial X_n} [X_n^{(1)} - X_n^{(0)}] \quad (2-20)$$

or in general

$$\epsilon_j^{(m)} = - \sum_{n=1}^p \frac{\partial \phi_j^{(m)}}{\partial X_n} \Delta X_n^{(m)} \quad (2-21)$$

where

$$\Delta X_n^{(m)} = X_n^{(m+1)} - X_n^{(m)}.$$

Eq. (2-21) represents a set of algebraic equations in which the unknowns are  $\Delta X_n^{(m)}$ . The partial derivatives,  $\partial \phi_j^{(m)} / \partial X_n$ , which can be calculated analytically or by finite difference, represent known coefficients in the equation. Because the indices  $j$  and  $n$  both vary from 1 to  $p$ , this set of equations may be seen to be linear algebraic equations with  $p$  equations and  $p$  unknowns. By means of such an approach as Cramer's Rule, successive elimination, or iteration, the values of the  $\Delta X_n^{(m)}$  can be established.

In actual practice, the repeated application of the process will result in a systematic reduction of the residuals toward zero, if convergence occurs. Normally, a solution is considered to have been obtained when all residuals have been reduced to some prescribed level.

Convergence criteria for the Newton-Raphson method can be given by means of a Lipschitz condition as discussed in reference 6. Unfortunately such criteria inherently require a knowledge of the solution before providing an indication of convergence. Thus, for practical purposes, application of the criteria is of little value. The most useful tool actually available in this respect is the Jacobian matrix

$$[\partial \phi_j / \partial X_n] .$$

If the determinant of this matrix is identically zero, then the set of equations represented by Eq. (2-21) is not linearly independent, and there is no unique solution to this set of equations. For this case the Newton-Raphson method fails. If the determinant of the matrix for any set of estimates  $X_n^{(m)}$  numerically



approaches zero or is very small, the Newton-Raphson method tends to diverge. In such a situation, a different set of estimates with a non-zero determinant must be used and the process restarted.

Based on the brief discussion presented, it is obvious that an awareness of the value of determinant of the Jacobian matrix is of importance in using the Newton-Raphson method. The most practical method of establishing whether or not convergence will occur, however, is the actual application of the technique. For most cases, based on practical experience, if a solution has not been obtained before  $m$  reaches some limiting value  $U_{\text{limit}}$ , (usually set at 20), convergence is not likely to occur.

### 2.3.2 Method of Steepest Descent

Several different versions of this method are available as discussed in references 6 and 9. For the sake of brevity, only that method which was actually used in the present study will be described.

In its most general applications, the method applies to the location of the maximums or minimums of a function of several real variables. Thus, if the method is to be employed in the problem under consideration, the set of  $p$  equations must be represented by some function with a minimum point corresponding to each set of roots to the equations.

The function

$$\tau(X_1, X_2, \dots, X_p) = \sum_{j=1}^p (\phi_j)^2 \quad (2-22)$$

can be seen to meet these requirements.

The equation

$$\tau(X_1, X_2, \dots, X_p) = \text{constant} \quad (2-23)$$

represents a surface in p-dimensional space. From vector analysis, at any point  $(X_1^{(m)} \dots X_p^{(m)})$ , the gradient of  $\tau$ ,  $\nabla\tau$ , is a vector which is normal to the surface represented by Eq. (2-23). For this reason, the vector  $(-\nabla\tau)$  points in the direction of most rapid decrease of  $\tau$  with respect to the point  $(X_1^{(m)} \dots X_p^{(m)})$ . Initially, as the vector is extended from the point  $(X_1^{(m)} \dots X_p^{(m)})$ , it intersects surfaces with successively smaller values of  $\tau$ . Because  $\tau$  can only have positive values for real variables, the vector  $-\nabla\tau(X_1^{(m)} \dots X_p^{(m)})$  will eventually intersect a surface with a value of  $\tau$  which is smaller than the value of  $\tau$  for any other surface which the vector will intersect. If the vector is extended beyond this point of intersection, the surfaces intersected will have successively larger values of  $\tau$ . If the point  $(X_1^{(m)} \dots X_p^{(m)})$  is represented by the position vector  $\vec{X}^{(m)}$ ,  $\tau$  can be treated as a function of  $\vec{X}^{(m)}$ . The point of intersection of the vector  $[-\nabla\tau(\vec{X}^{(m)})]$  with the surface of minimum value of  $\tau$  can thus be expressed by the position vector

$$\vec{X}^{(m+1)} = \vec{X}^{(m)} - \lambda \nabla\tau(\vec{X}^{(m)}) \quad (2-24)$$

where  $\lambda$  is a variable whose magnitude is not yet known. The value of  $\lambda$  can be established by treating  $\tau$  as a function of  $\lambda$  as follows:

$$\begin{aligned} \tau(\vec{X}^{(m+1)}) &= \tau[\vec{X}^{(m)} - \lambda \nabla\tau(\vec{X}^{(m)})] \\ &= \tau^{(m+1)}(\lambda) \end{aligned} \quad (2-25)$$

From basic calculus, when  $\tau^{(m+1)}(\lambda)$  is at a minimum (or a maximum) with respect to the independent variable  $\lambda$ ,

$$\frac{d \tau^{(m+1)}(\lambda)}{d \lambda} = 0 \quad (2-26)$$

This differential can be expanded as follows:

$$\begin{aligned} \frac{d \tau^{(m+1)}(\lambda)}{d \lambda} &= \frac{\partial \tau^{(m+1)}}{\partial X_1} \frac{dX_1}{d \lambda} + \frac{\partial \tau^{(m+1)}}{\partial X_2} \frac{dX_2}{d \lambda} + \dots + \frac{\partial \tau^{(m+1)}}{\partial X_p} \frac{dX_p}{d \lambda} \\ &= \sum_{n=1}^p \frac{\partial \tau^{(m+1)}}{\partial X_n} \left( - \frac{\partial \tau^{(m)}}{\partial X_n} \right) \\ &= - [\nabla \tau(\vec{X}^{(m+1)}) \cdot \nabla \tau(\vec{X}^{(m)})] \end{aligned} \quad (2-27)$$

Thus  $\lambda$  must satisfy the relation

$$\nabla \tau(\vec{X}^{(m)}) \cdot \nabla \tau[\vec{X}^{(m)} - \lambda \nabla \tau(\vec{X}^{(m)})] = 0 \quad (2-28)$$

Equation (2-28) must be solved by some iterative scheme because it is itself a nonlinear algebraic equation with a single unknown,  $\lambda$ . In practice it is convenient to approximate  $\lambda$  in a single step,

$$\lambda \approx \nabla \tau(\vec{X}^{(m)}) \cdot \nabla \tau(\vec{X}^{(m)}) / \left[ \nabla \tau(\vec{X}^{(m)}) \cdot \frac{d \nabla \tau^{(m)}(\lambda)}{d \lambda} \right] \quad (2-29)$$

This in essence represents an application of the Newton-Raphson method to obtain a value for  $\lambda$ .

Once  $\lambda$  is calculated, the new position vector  $X^{(m+1)}$  is established. The process is repeated until the value of each  $\Delta X_n^{(m)}$ , where

$$\Delta X_n^{(m)} = -\lambda \frac{\partial \tau}{\partial X_n} (\vec{X}^{(m)}) \quad (2-30)$$

is less than some prescribed limit. At this point the values of  $X_n^{(m)}$  must be examined to determine whether or not they represent roots to the set of equations. In some cases the method converges to values of  $X_n$  which represent a "false minimum" such that the value of  $\tau$  is not close to zero. For such a case the values of  $X_n$  obtained do not represent roots to the equations.

### 2.3.3 Freudenstein-Roth Technique

In applying either the Newton-Raphson method or the Method of Steepest Descent, convergence is very likely not to occur unless the initial estimates of the roots are in the neighborhood of the actual values. Obviously, in many cases, the locations of such neighborhoods are unknown. The need therefore arises for some means of improving initial estimates of the roots.

One method which shows promise is the Freudenstein-Roth technique as described in reference 3. Because of its flexibility, many different variations of this approach can be developed. Only the two versions actually used will be discussed here. For brevity the first of these will be referred to as the "coefficient approach" and the second as the "constant approach".

The first step in the coefficient approach involves assuming some set of initial values,  $X_n^{(0)}$ , for the roots. These initial values will generally not satisfy the original equations. They do, however, represent a set of roots to the following set of equations:

$$A_j^{(0)} t_{j1}^{(0)} + \sum_{i=2}^{Q_j} t_{ji}^{(0)} = F_j \quad (j=1,2,\dots,p) \quad (2-31)$$

where

$$A_j^{(0)} = (F_j - \phi_j^{(0)})/t_{j1}^{(0)} + 1 \quad (2-32)$$

and

$$t_{ji}^{(m)} = \prod_{k=1}^{d_j} Y_{r(j,i,k)}^{(m)}$$

Eq. (2-31) represents a set of equations which differs from the set represented by Eq. (2-15) only in that the coefficients  $A_j^{(0)}$  appear with the first term,  $t_{j1}$ .

After the values of the coefficients have been determined, a recursion relation, to cause each  $A_j$  to approach unity, is established. This relation can be based on several alternate methods. For a simple linear variation of  $A_j^{(m)}$ , the following relation is sufficient:

$$A_j^{(m)} = A_j^{(0)} - \frac{m(F_j - \phi_j^{(0)})}{t_{j1} V} \quad (m=0,1,2,\dots,V) \quad (2-33)$$

where

$V$  = the selected number of iterative steps in the Freudenstein-Roth technique.

However, when

$$\left| \frac{F_j - \phi_j^{(0)}}{t_{j1}} \right| \gg 1 \quad (2-34)$$

the following logarithmic variation is more useful:

$$A_j^{(m)} = \left[ A_j^{(0)} \right]^{\frac{V-m}{V}} \quad (m=0,1,2\dots V) \quad (2-35)$$

By means of either recursion relation, the values of  $A_j^{(1)}$  may be calculated, and a new set of equations established corresponding to Eq. (2-31). The values of  $X_n^{(0)}$  are then used as initial estimates for the roots of the new set of equations. If  $A_j^{(1)}$  are sufficiently close to  $A_j^{(0)}$ , these initial estimates will be in the neighborhood of the actual roots,  $X_n^{(1)}$ , for the new set of equations. Thus by means of the Newton-Raphson method, the actual roots,  $X_n^{(1)}$ , can be calculated for the set of equations containing  $A_j^{(1)}$ . After such calculations, by means of the recurrence formula  $A_j^{(2)}$  are calculated.  $X_n^{(1)}$  are then used as initial estimates for calculating  $X_n^{(2)}$ . The process is repeated until on the  $V^{\text{th}}$  step, the coefficients  $A_j^{(V)}$  equal unity and the initial estimates  $X_n^{(V-1)}$  are used to calculate  $X_n^{(V)}$  which represent the true roots to Eq. (2-15).

In the constant method, the procedure is similar to that for the coefficient method except rather than varying the coefficients, the constant terms,  $F_j$ , are varied. Based on a set of initial estimates,  $X_n^{(0)}$ , the values  $F_j^{(0)}$  are established by the relation

$$F_j^{(0)} = \phi_j^{(0)} \quad (j=1,2\dots p) \quad (2-36)$$

Based on the linear recursion relation

$$F_j^{(m)} = F_j^{(0)} + \frac{m(F_j^{(V)} - F_j^{(0)})}{V} \quad (m=0,1,2\dots V) \quad (2-37)$$

where

$$F_j^{(V)} = F_j \text{ (the true value of the constant term)}$$

the values of  $F_j^{(m)}$  are systematically changed. With each such change, a new set of roots ( $X_n^{(m)}$  for the corresponding set of equations) are calculated based on the preceding roots  $X_n^{(m-1)}$ , and using the Newton-Raphson method. As before, in the  $V^{\text{th}}$  step, the value of  $X_n^{(V-1)}$  is used as the initial estimate of  $X_n^{(V)}$  which represents the set of true roots to Eq. (2-15).

The relative performance of the coefficient and constant approach is not clearly established. However, the constant method appears superior to the coefficient method whenever the initial set of coefficients,  $A_j^{(0)}$ , used in the latter method, are negative. To avoid such negative coefficients when using the coefficient method, it is desirable to arrange the terms  $t_{ji}$  in each equation so that for a given set of initial values,  $X_n^{(0)}$ , the term with the greatest magnitude is  $t_{j1}$ .

The convergence criteria for the Freudenstein-Roth method are discussed in reference 3. Obviously, if the step size is large, reflecting a small value of  $V$ , the Newton-Raphson method may fail. This problem can be corrected by increasing the value of  $V$ , but a point may be reached beyond which further increases in  $V$  are not practical. In such a case, the problem must be started over using a new set of estimates.

## 2.4 Selection of Initial Estimates

In the problem under consideration, the range of values, within which the desired roots are found, is quite large as indicated by Table 2-1. Under such conditions, if the individual roots within a set are of different orders of magnitude, considerable difficulty may be encountered in achieving convergence with any of the numerical methods described. Thus, the need arises for some additional tool to provide the best possible initial estimates. One such tool which was developed in the course of the investigation appears worthy of discussion.

The first step in the procedure involves establishing a "range of interest" for each unknown. This range must equal or fall within the range of the appropriate electrical element as provided in Table 2-1. Based on this range of interest, recursion relations and mean value relations are established as follows:

$$Y_n^{(m)} = Y_{n(\text{low})} \cdot [Y_{n(\text{up})}/Y_{n(\text{low})}]^{\frac{m-1}{W-1}} \quad (m=1, 2, \dots, W) \quad (2-38)$$

$$\bar{Y}_n = [Y_{n(\text{low})} \cdot Y_{n(\text{up})}]^{\frac{1}{2}} \quad (2-39)$$

where

$Y_{n(\text{low})}$  = lower limit of the range of interest for  $Y_n$

$Y_{n(\text{up})}$  = upper limit of the range of interest for  $Y_n$

$W$  = number of steps selected for use with the process.



The values of  $Y_1^{(1)}$  and the values  $\bar{Y}_2, \bar{Y}_3 \dots \bar{Y}_p$  are used to calculate  $\psi_j^{(1,1)}$  and to determine the sum of the absolute values of the corresponding residuals,

$$\sum_{j=1}^p |\psi_j^{(1,1)} - F_j|.$$

$Y_1^{(m)}$  is then varied according to Eq. (2-38) while  $\bar{Y}_2, \dots, \bar{Y}_p$  are held constant. The sum of the absolute values of the residuals resulting from each step are compared to determine the minimum sum. The value of  $Y_1^{(m)}$ , corresponding to the minimum value of

$$\sum_{j=1}^p |\psi_j^{(m,1)} - F_j|,$$

is selected and its superscript designated (a11). The value  $Y_1^{(a11)}$  is then used with  $Y_2^{(1)}$  and  $\bar{Y}_3 \dots \bar{Y}_p$  to calculate  $\psi_j^{(1,2)}$  and the corresponding sum of residuals. With  $\bar{Y}_1^{(a11)}$  and  $\bar{Y}_3 \dots \bar{Y}_p$  held constant,  $Y_2^{(m)}$  is varied according to Eq. (2-38) until a minimum sum is established. The corresponding  $Y_2^{(m)}$  is designated  $Y_2^{(a12)}$  and is then used along with  $Y_1^{(a11)}, Y_3^{(m)}, \bar{Y}_4 \dots \bar{Y}_p$  to find  $Y_3^{(a13)}$  etc. After p cycles the values of  $Y_1^{(a11)} \dots Y_p^{(alp)}$  are established. At this point the process starts over using  $Y_1^{(1)}$  and  $Y_2^{(a12)} \dots Y_p^{(alp)}$  as initial values. The value of the superscript of  $Y_1^{(m)}$ , corresponding to the minimum sum of the absolute value of the residuals, is established and designated a21. This superscript is compared with a11. If  $|a11 - a21| \leq \Delta a$ , where  $\Delta a$  is some selected integer, the process continues using  $Y_1^{(a11)}, Y_2^{(1)}, Y_3^{(a13)} \dots Y_p^{(alp)}$  as initial values to establish  $Y_2^{(a22)}$  etc. The process is considered complete if for p consecutive cycles,

$$|a1n - a2n| \leq \Delta a.$$

If at any point,

$$|a_{1n} - a_{2n}| > \Delta a,$$

the old value of  $Y_n^{(a_{1n})}$  is discarded and  $Y_n^{(a_{1n})}$  is set equal to  $Y_n^{(a_{2n})}$ .

Following such an event the process must continue on for at least  $p$  cycles without  $\Delta a$  being exceeded by  $|a_{1n} - a_{2n}|$ . Whenever  $\Delta a$  is exceeded, the counting starts over until  $p$  consecutive cycles are achieved without an inequality. The resulting estimates of  $Y_n$ , designated  $Y_n^{(g)}$ , may then be used as an estimate in conjunction with the other numerical techniques already described.

## 2.5 Digital Computer Considerations

The numerical procedures previously outlined are especially suited for use with a digital computer. The most important factor to consider in developing a computer program involving such techniques is general applicability to different sets of equations. Thus the program should be flexible enough to solve any set of equations of the form of Eq. (2-2) irregardless of the number of equations or unknowns ( $p$ ), the number of terms in each equation ( $Q_j$ ), or the number of unknowns in each term ( $d_j$ ).

To achieve the desired flexibility, the identity of each unknown in each term of each equation must be an input. The method developed for accomplishing this arrangement is based on the concept of assigning subscripts to the subscripts as indicated by Eq. (2-4). Based on the relationship between circuit elements and unknowns established by Eqs. (2-5), (2-7), and (2-8) a numerical value (1,2,...n...p) can be assigned for each subscripted subscript  $r(j,i,k)$ . The values for  $r(j,i,k)$  can thus be read into the computer and stored in a

systematic manner. For example the subscript  $r(3,4,2)$  refers to the second factor ( $k=2$ ) of the fourth term ( $i=4$ ) in the third equation ( $j=3$ ). If the subscripted variable  $Y_{r(j,i,k)}$  is called for, the computer determines the value,  $n$ , which was the input for  $r(j,i,k)$ . With the value of the subscript established, the appropriate  $Y_n$  which has either been input or calculated, is used for  $Y_{r(j,i,k)}$ . For example, if  $r(3,4,2) = 6$ , then  $Y_{r(3,4,2)} = Y_6$ .

Because the subscript of a particular unknown  $Y_n$  is identical to the subscript of the corresponding  $X_n$ , the same technique for subscripted subscripts described for  $Y_n$ , applies also to  $X_n$ . Based on practical considerations, it appears most convenient to establish the equations in terms of the unknown variables  $Y_n$  and then to convert the equations by a change of variables from  $Y_n$  to  $X_n$  according to Eq. (2-14) within the computer program. Thus all inputs would refer to the variables  $Y_n$ . Likewise, all outputs are most conveniently expressed in terms of  $Y_n$ .

### 3.0 DIGITAL COMPUTER PROGRAM

Based on the technical considerations described in Section 2.0, a digital computer program has been developed in both FORTRAN II and FORTRAN IV for the simultaneous solution of sets of nonlinear algebraic equations. The program is designed as a working tool with maximum flexibility built-in. The subsections which follow provide a description of the various features of the program.

#### 3.1 Basic Features

The program is designed to solve sets of nonlinear algebraic equations of the type indicated by Eq. (2-2). A general flow chart is provided in Figure 3-1. Copies of the source program written in FORTRAN II and IV are included in Appendix A. A description of the subroutines included in the program is presented in Appendix B. This program has been checked out for use on the IBM 7094 digital computer in FORTRAN IV and for the CDC 3200, SDS 930, and IBM 1620 computers in FORTRAN II. For the case involving more than 10 equations and 10 unknowns, the core storage of the SDS 930 will probably be exceeded.

The program utilizes the Freudenstein-Roth technique in conjunction with the Newton-Raphson method. For the variation of the coefficients in the Freudenstein-Roth technique, the logarithmic variation as expressed by Eq. (2-35) is used. For the Newton-Raphson method all partial derivatives are calculated by analytical differentiation as opposed to finite difference methods. The Gaussian pivotal technique is used to solve the linear algebraic equations shown in Eq. (2-21).

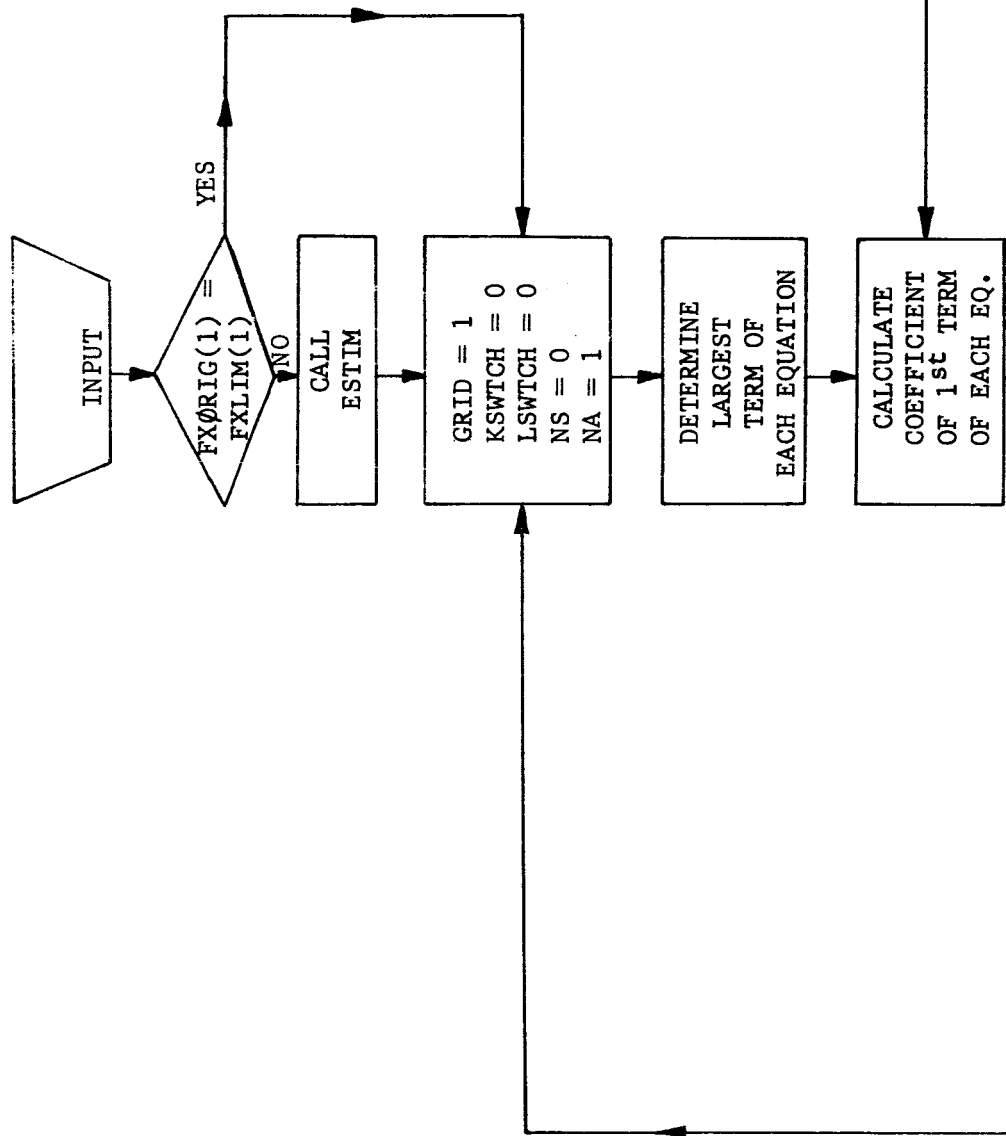
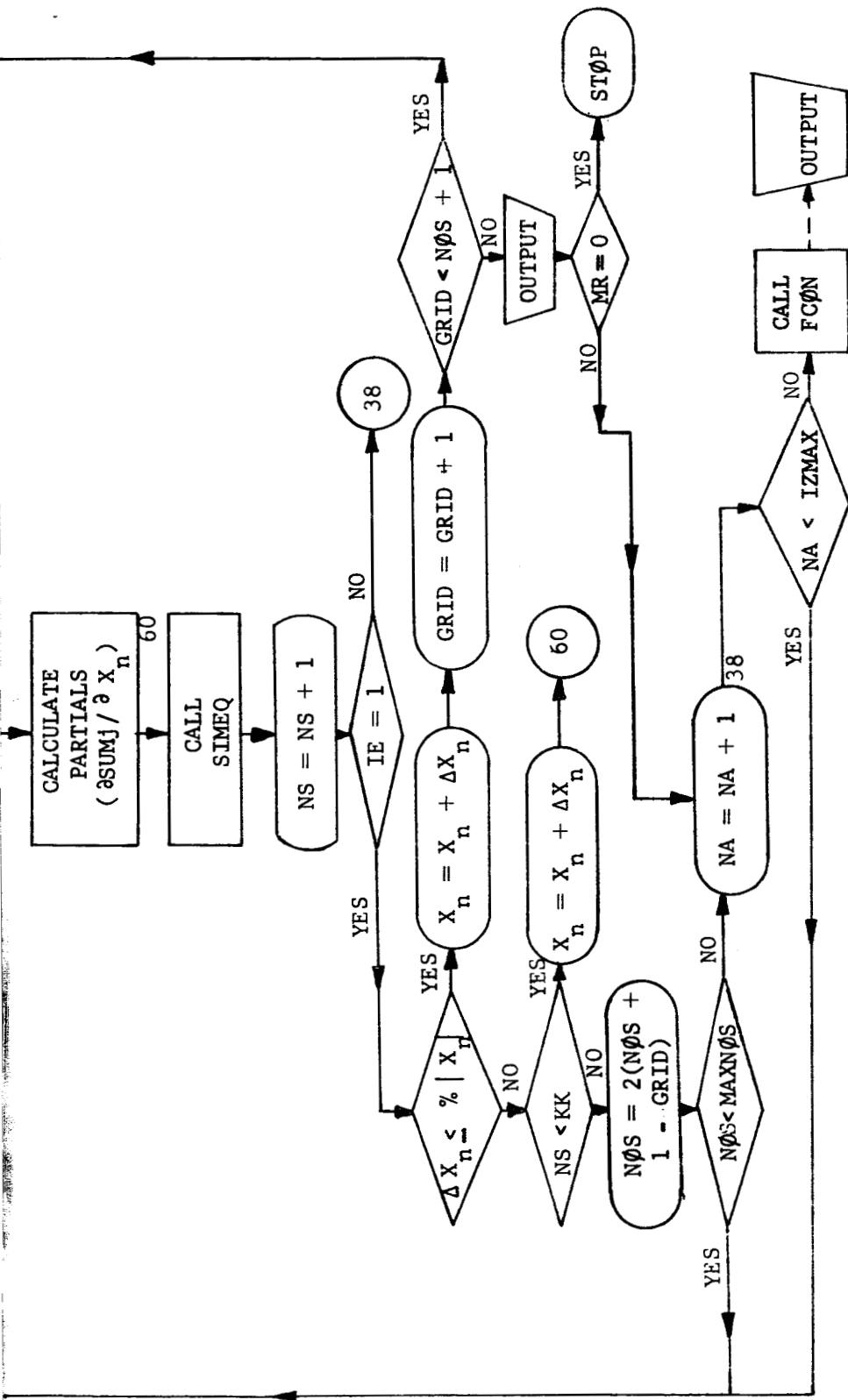


Figure 3-1a. PROGRAM FOR SOLUTION OF



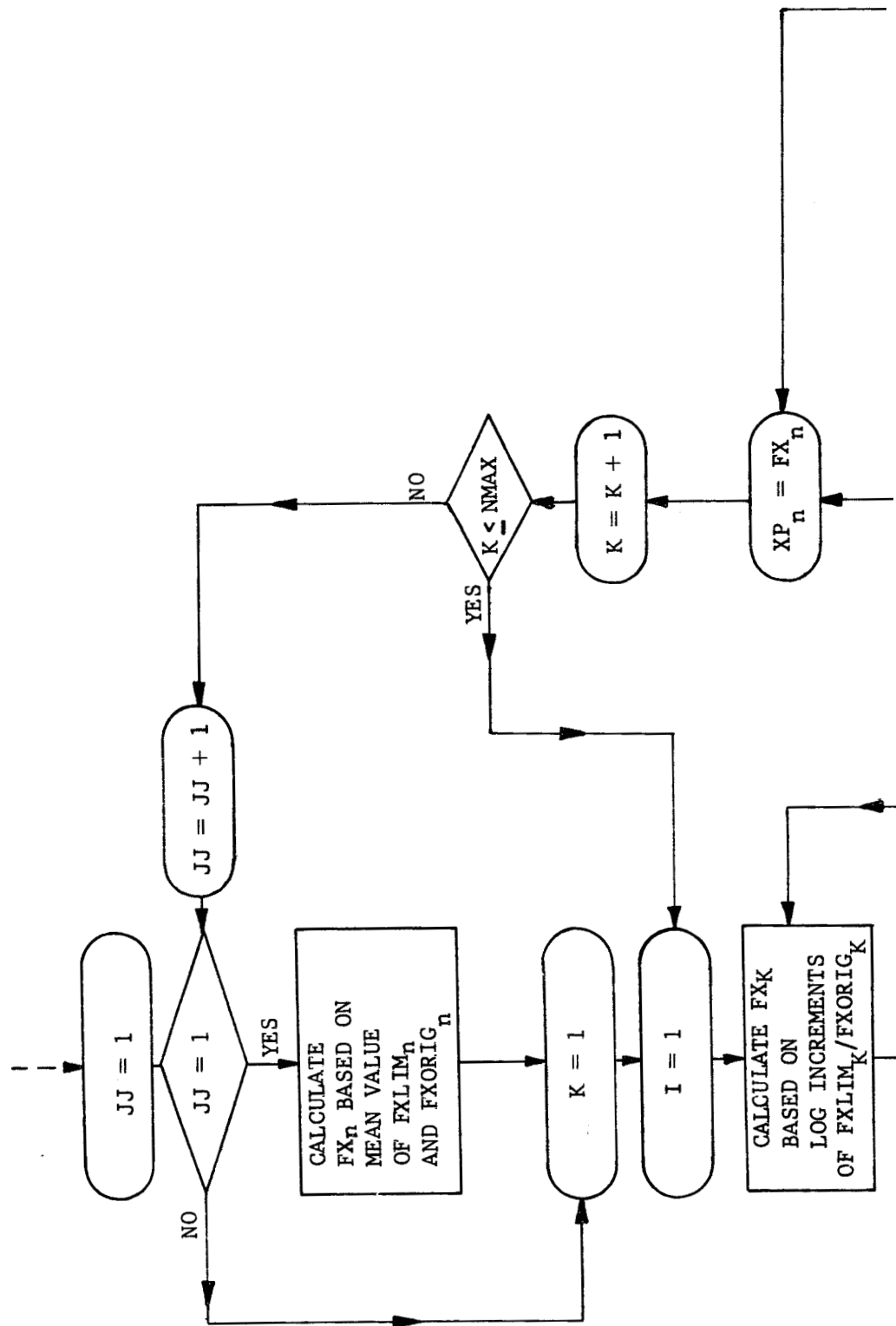
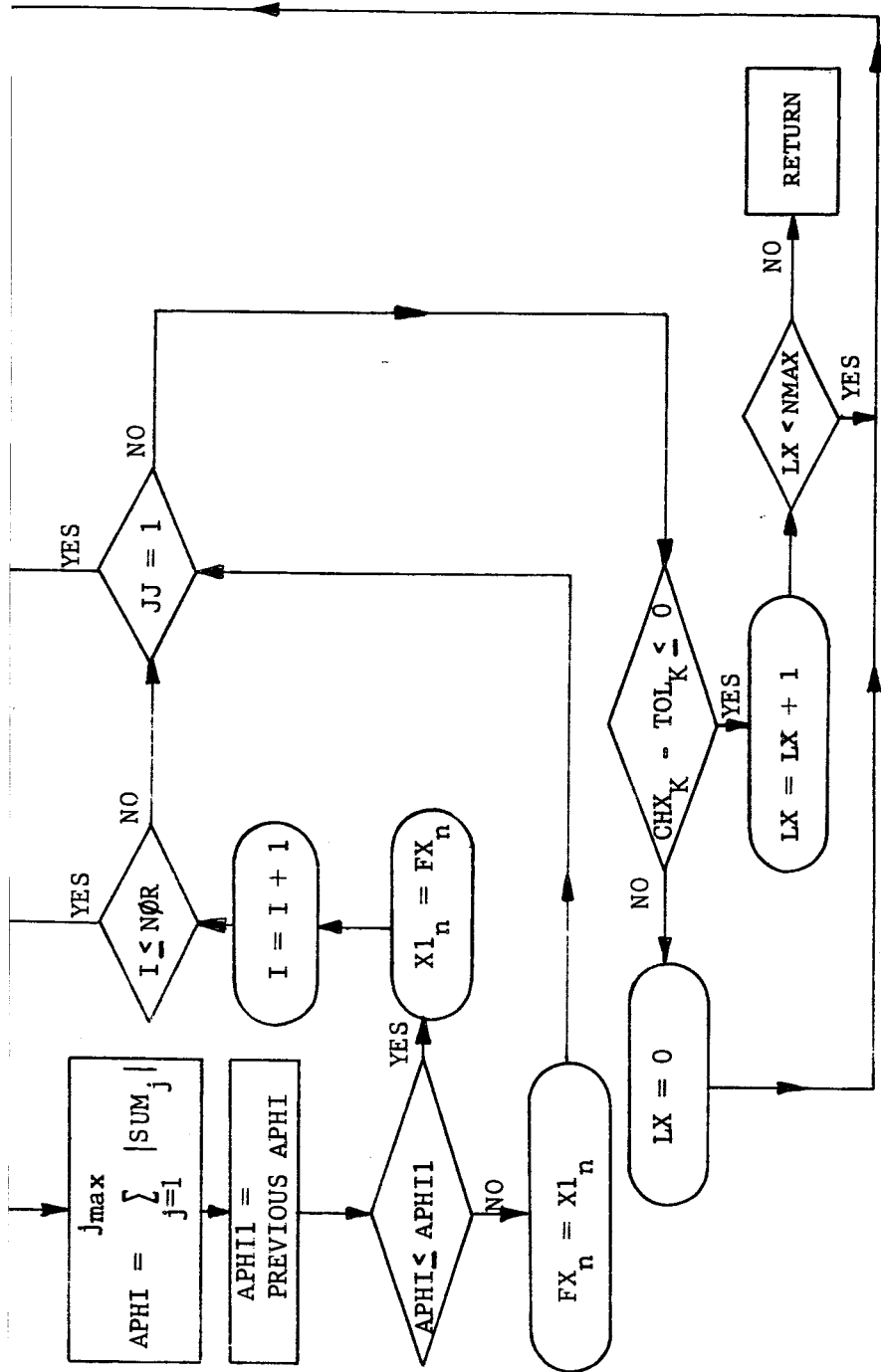


Figure 3-1b. SUBROU

2





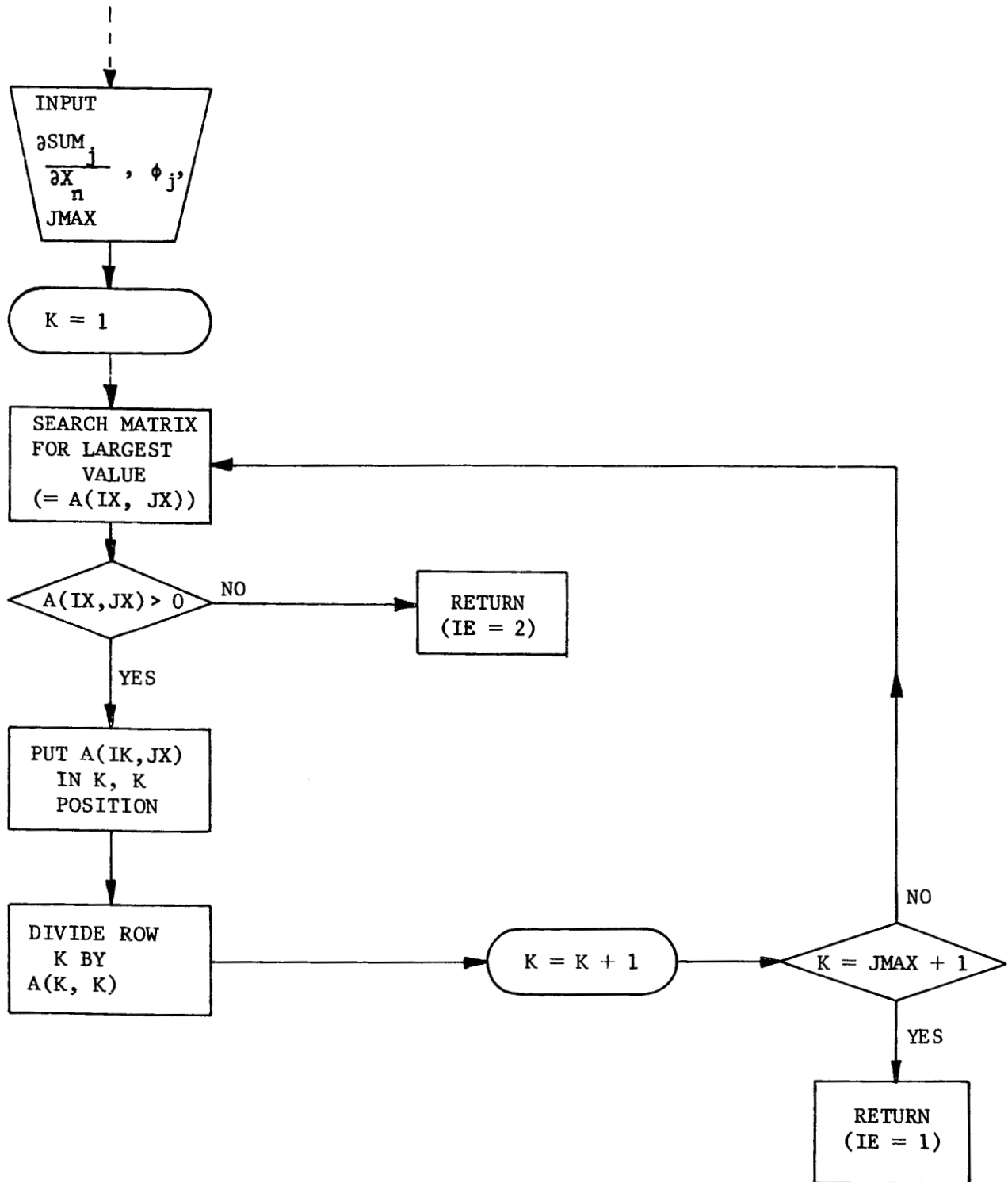


Figure 3-1c. SUBROUTINE SIMEQ

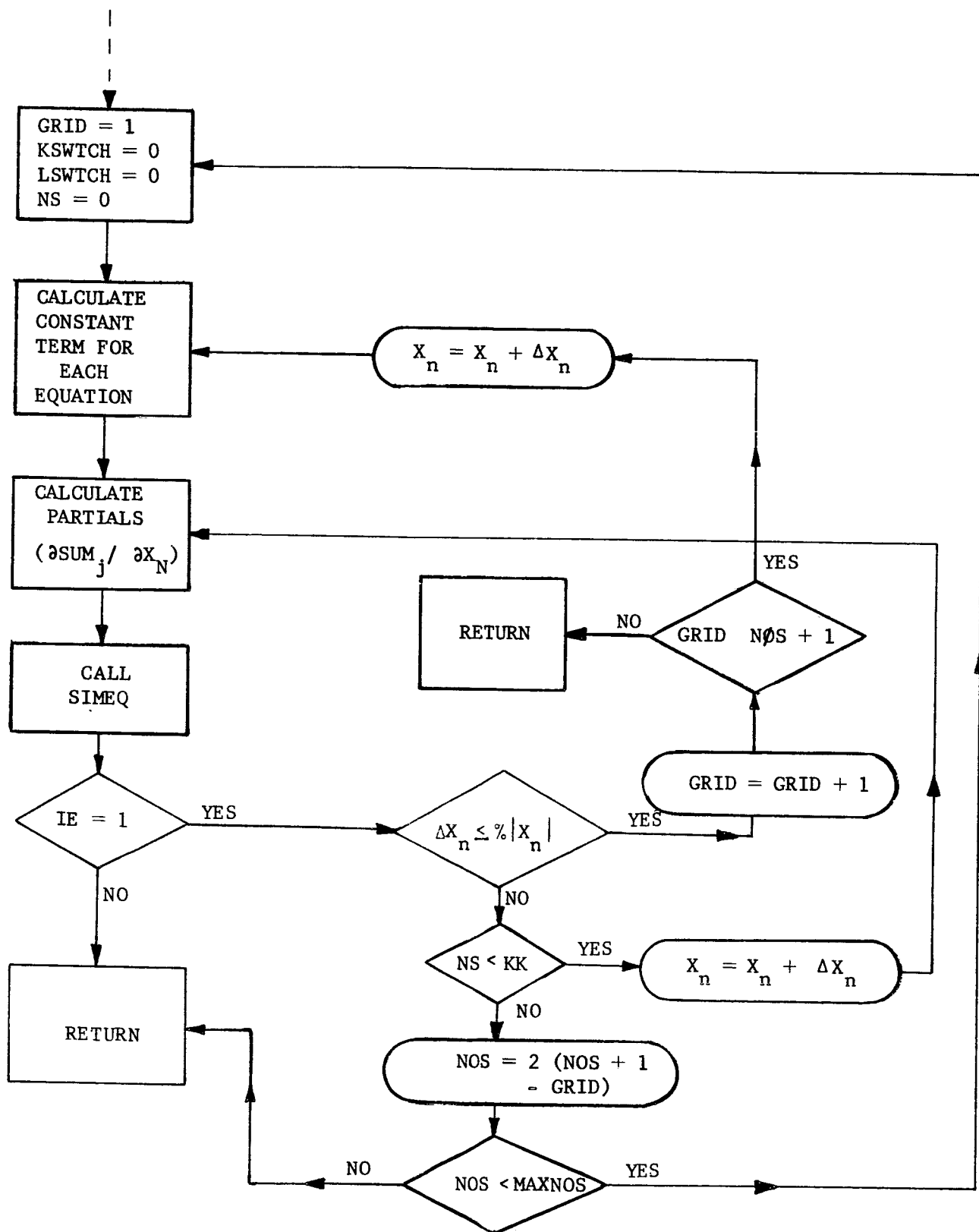


Figure 3-1d. SUBROUTINE FCON

After the terms of the equations and the initial range of interest for each unknown have been read into the computer, values for initial estimates of each unknown are established by means of the method for selection of initial estimates, discussed in Section 2.4. The terms are then rearranged so that the first term in each equation is that term with the largest numerical value. With such an arrangement of terms, and with the initial estimates, a solution is attempted using the coefficient method described in Section 2.3. If such a solution is obtained, it is printed out. If this is not successful, the value of  $V$  (the selected number of iterative steps) is doubled to reduce the size of the incremental change in the coefficients, and a solution is again attempted. If no solution is achieved,  $V$  is again doubled. The process continues until a solution is found or the value of  $V$  exceeds some established limit,  $V_{\text{limit}}$ . At this point the terms are rearranged in their original order (the order in which they were read into the computer) and a solution is again attempted. If this fails, the first column of terms is replaced by the second column and again an attempt is made to obtain a solution. The procedure is repeated as necessary, using the coefficient approach, with the first column of terms being successively replaced by the other columns within the set of equations. The total number of such attempts is equal to some value,  $Q_{\text{limit}}$ , which is equal to or less than the number of terms in the longest equation plus one ( $Q_{j(\text{max})} + 1$ ). In those equation where  $Q_j < Q_{j(\text{max})}$ , after all terms have been successively used as the first term, the term which was originally last is used repeatedly as the first term for all additional attempts.

If the coefficient approach does not yield a solution, the program, through a subroutine, applies the constant approach. This approach uses the same value of  $V$  as was initially used with the coefficient approach. The value,  $V_{\text{limit}}$ , is also used as before. If the constant approach fails, a new set of initial estimates must be calculated by, or read into the computer and the entire procedure repeated.

### 3.2 Inputs and Options

A complete listing of all items to be input is included in Appendix C. This listing includes FORTRAN symbol definitions and units corresponding to each input item. Input items include the values of the constants,  $F_j$ ; the range of interest for each unknown,  $Y_{n(\text{low})}$ , to  $Y_{n(\text{up})}$ ; and the subscripts identifying the arrangement of the unknowns in each equation,  $r(j,i,k)$ .

Extreme care should be taken in preparing the values of the subscripts of the unknowns to be used as inputs. The correspondence between the subscripts of the unknowns,  $Y_n$ , and the resistances, inductances, and capacitances should be in accordance with Eqs. (2-5), (2-7), and (2-8). If this correspondence is not used, or if any individual subscript is not correctly read into the computer, the set of equations which the computer will solve will not correspond to the actual set. Thus the solution obtained will not satisfy the correct set of equations.

An option is provided for the manner in which the initial set of estimates are established. If the values of  $Y_{n(\text{up})}$  are read in as equal to  $Y_{n(\text{low})}$ , the computer will use the value  $Y_{n(\text{low})}$  as the initial estimates without use of the method of selection of initial estimates

previously described. Otherwise, the computer will use the input values of  $Y_{n(\text{low})}$  and  $Y_{n(\text{up})}$  in conjunction with the method of selecting initial estimates as described in Section 2.4. Notice should be taken that for convenience the values of  $Y_n$ , where  $(u+v) < n \leq p$ , are read in as capacitances, and not as the reciprocals of the capacitances. The reciprocals for these  $Y_n$  are taken within the program.

Options are also provided for establishing the limiting value,  $V_{\text{limit}}$ , for the number of iterative steps in the Freudenstein-Roth technique, and also for  $U_{\text{limit}}$  which is the maximum number of steps to be used in the Newton-Raphson method.

The input item,  $K_m$ , governs the linear relationships between an inductance and its associated natural resistance in accordance with Eq. (2-11). Normally the input values of  $K_m$  are all unity, which is compatible with accepted electrical engineering practice. For those resistances which are not associated with any inductance, the value of  $K_m$  should be input as zero.

An option is established as to the number of columns of terms to be interchanged in the manner already described in subsection 3.1. The input value of  $Q_{\text{limit}}$  establishes the number of such exchanges. There is also an option as to what action the computer takes after it obtains a set of satisfactory roots. Depending on the value of MR (FORTRAN symbol) the computer may stop after finding a set of roots or may continue searching for additional sets of roots until all limits ( $U_{\text{limit}}$ ,  $V_{\text{limit}}$ , and  $Q_{\text{limit}}$ ) have been reached.

### 3.3 Outputs

The outputs from the program are designed to provide the user not only with the solution to the problem but also with an indication of the steps taken in obtaining a solution. A complete description of all outputs is contained in Appendix C which includes a sample printout.

The printout first provides a record of important input data and then gives a running account of progress toward a solution. Included in this account are a statement as to which Freudenstein-Roth technique (coefficient or constant) is being applied; the column number of the terms which have been exchanged with the first column (applicable only to the coefficient approach); and the iterative step number versus the total number of steps in use with the Freudenstein-Roth technique. When a solution is obtained, the computer indicates whether or not the roots lie within the physical limits specified. Following this statement, the roots, written in terms of resistances, inductances, and capacitances are listed complete with units. If no roots are found, a statement indicating this fact is printed out.

#### 4.0 DISCUSSION OF RESULTS

The computer program described in Section 3.0 was used to solve several different sets of nonlinear equations of the form of Eq. (2-2). In addition to the program in the form presented, several other versions of the program were tested. The results obtained are described in the subsections which follow.

##### 4.1 Relative Performance of Numerical Methods

One version of the computer program contained a subroutine which used the Method of Steepest Descent, as described in Section 2.3.2, to solve the equations. This subroutine did not obtain solutions unless the initial guesses were extremely close to the true roots. The Freudenstein-Roth technique combined with the Newton-Raphson method consistently solved sets of equations which the Method of Steepest Descent, with the same set of initial estimates, could not solve. The inferior performance of the latter method apparently resulted from a tendency toward convergence to false minima, the coordinates of which did not represent true roots to the equations. This undesirable tendency was possibly due to the considerable variation in the orders of magnitude of the unknowns which caused the residual of some equations to outweigh greatly the residuals of others. Because of the lack of success with the Method of Steepest Descent, the subroutine employing this technique was deleted from the program in its final form.

Because of the variation in the orders of magnitude of the unknowns, the need arose for some method which could possibly improve on an initial

set of estimates. Accordingly the method described in Section 2.4 was incorporated into the program. The use of this technique proved beneficial in some cases, but due to time limitations its true value has not yet been established.

#### 4.2 Application to Actual Problems

The digital computer program was used to solve sets of six equations and thirteen equations. The equations represented filter circuits described in ref. 10. A set of fifteen equations based on a filter circuit described in ref. 11 was also developed, but time limitations have not permitted application of the digital computer program to this set.

##### 4.2.1 Six Equations

The program successfully solved a set of six equations and six unknowns which represent the filter circuit on page B-42 of ref. 10. The equations are included in Appendix D. Table 4-1 presents representative values of  $Y_{n(\text{low})}$  and  $Y_{n(\text{up})}$  used and the roots obtained. In general the program experienced no difficulty handling this set of equations.

##### 4.2.2 Thirteen Equations

Solution of a set of thirteen equations and thirteen unknowns, based on the filter circuit shown on page B-93 of ref. 10 was also attempted by the computer program. The equations are presented in Appendix E. Table 4-2 presents the values of  $Y_{n(\text{low})}$  and  $Y_{n(\text{up})}$  used and the resulting roots.



TABLE 4-1

## RESULTS OF COMPUTER SOLUTION OF SIX EQUATIONS

<u>Run #</u>	<u>n</u>	<u>Y<sub>n(low)</sub></u>	<u>Y<sub>n(up)</sub></u>	<u>Y<sub>n</sub></u>	<u>Corresponding Electrical Components</u>
L(6-4-F)	1	4,000	4,000	2535.5886	R(1)=2.5355886 x 10 <sup>3</sup> ohms
	2	6,000	6,000	2136.2946	R(2)=2.1362946 x 10 <sup>3</sup> ohms
	3	800	800	561.72030	R(3)=5.6172030 x 10 <sup>2</sup> ohms
	4	40	40	28.086015	L(1)=2.8086015 x 10 <sup>1</sup> henries
	5	80	80	28.483928	L(2)=2.8483928 x 10 <sup>1</sup> henries
	6	20,000	20,000	12,601.871	C(1)=7.93532960 x 10 <sup>-5</sup> farads
B(6-8-FC)	1	1,500	1,500	1,999.9954	R(1)=1.9999954 x 10 <sup>3</sup> ohms
	2	2,500	2,500	3,000.0089	R(2)=3.0000089 x 10 <sup>3</sup> ohms
	3	250	250	399.99882	R(3)=3.9999882 x 10 <sup>2</sup> ohms
	4	15	15	19.999939	L(1)=1.9999939 x 10 <sup>1</sup> henries
	5	43	43	40.00017	L(2)=4.0000117 x 10 <sup>1</sup> henries
	6	8,000	8,000	9,999.9774	C(1)=1.00000226 x 10 <sup>-4</sup> farads

TABLE 4-2

## RESULTS OF COMPUTER SOLUTION OF THIRTEEN EQUATIONS

<u>Run #</u>	<u>n</u>	<u><math>Y_{n(low)}</math></u>	<u><math>Y_{n(up)}</math></u>	<u><math>Y_n</math></u>	<u>Corresponding Electrical Component</u>
3200(J-12)	1	$1.80 \times 10^3$	$1.00 \times 10^4$	$3.16134118 \times 10^3$	$R(1) = 3.16134118 \times 10^3$ ohms
	2	$3.60 \times 10^3$	$1.00 \times 10^4$	$1.06913525 \times 10^4$	$R(2) = 1.06913525 \times 10^4$ ohms
	3	$4.50 \times 10^3$	$1.00 \times 10^4$	$3.25326080 \times 10^3$	$R(3) = 3.25326080 \times 10^3$ ohms
	4	$2.70 \times 10^3$	$1.00 \times 10^4$	$1.12238574 \times 10^3$	$R(4) = 1.12238574 \times 10^3$ ohms
	5	$4.50 \times 10^2$	$1.00 \times 10^3$	$5.00 \times 10^2$	$R(5) = 5.00 \times 10^2$ ohms
	6	$4.50 \times 10^1$	$1.00 \times 10^2$	$8.60235493 \times 10^1$	$L(1) = 8.60235493 \times 10^1$ henries
	7	$5.40 \times 10^1$	$1.00 \times 10^2$	$1.60370290 \times 10^2$	$L(2) = 1.60370290 \times 10^2$ henries
	8	$3.60 \times 10^1$	$1.00 \times 10^2$	$2.42838392 \times 10^1$	$L(3) = 2.42838392 \times 10^1$ henries
	9	$2.70 \times 10^1$	$1.00 \times 10^2$	$1.1238573 \times 10^1$	$L(4) = 1.1238573 \times 10^1$ henries
	10	$9.00090009 \times 10^3$	$1.00 \times 10^2$	$4.2279636 \times 10^3$	$C(1) = 2.36520480 \times 10^{-4}$ farads
	11	$1.07991361 \times 10^4$	$1.00 \times 10^5$	$3.20740568 \times 10^4$	$C(2) = 3.11778458 \times 10^{-5}$ farads
	12	$1.35135135 \times 10^4$	$1.00 \times 10^5$	$2.93970045 \times 10^4$	$C(3) = 3.40170713 \times 10^{-5}$ farads
	13	$1.80180180 \times 10^4$	$1.00 \times 10^5$	$7.48257161 \times 10^2$	$C(4) = 1.33643893 \times 10^{-4}$ farads

TABLE 4-2 (Concluded)

## RESULTS OF COMPUTER SOLUTION OF THIRTEEN EQUATIONS

<u>Run #</u>	<u>n</u>	<u><math>Y_{n(low)}</math></u>	<u><math>Y_{n(up)}</math></u>	<u><math>Y_n</math></u>	<u>Corresponding Electrical Components</u>
3200(J-18)	1	$1.95 \times 10^3$	$1.00 \times 10^4$	$1.99999943 \times 10^3$	$R(1) = 1.99999943 \times 10^3$ ohms
	2	$3.94 \times 10^3$	$1.00 \times 10^4$	$3.99999984 \times 10^3$	$R(2) = 3.99999984 \times 10^3$ ohms
	3	$4.96 \times 10^3$	$1.00 \times 10^4$	$5.00000081 \times 10^3$	$R(3) = 5.00000081 \times 10^3$ ohms
	4	$2.97 \times 10^3$	$1.00 \times 10^4$	$3.00000012 \times 10^3$	$R(4) = 3.00000012 \times 10^3$ ohms
	5	$5.00 \times 10^2$	$1.00 \times 10^3$	$5.00 \times 10^2$	$R(5) = 5.00 \times 10^2$ ohms
	6	$5.00 \times 10^1$	$1.00 \times 10^2$	$4.99999885 \times 10^1$	$L(1) = 4.99999885 \times 10^1$ henries
	7	$6.00 \times 10^1$	$1.00 \times 10^2$	$5.99999977 \times 10^1$	$L(2) = 5.99999977 \times 10^1$ henries
	8	$4.00 \times 10^1$	$1.00 \times 10^2$	$4.00000098 \times 10^1$	$L(3) = 4.00000098 \times 10^1$ henries
	9	$3.00 \times 10^1$	$1.00 \times 10^2$	$3.00000012 \times 10^1$	$L(4) = 3.00000012 \times 10^1$ henries
	10	$1.00 \times 10^4$	$1.00 \times 10^2$	$1.00000031 \times 10^4$	$C(1) = 9.99999688 \times 10^{-5}$ farads
	11	$1.20048019 \times 10^4$	$1.00 \times 10^5$	$1.19999995 \times 10^4$	$C(2) = 8.33333366 \times 10^{-5}$ farads
	12	$1.49925037 \times 10^4$	$1.00 \times 10^5$	$1.49999942 \times 10^4$	$C(3) = 6.66666922 \times 10^{-5}$ farads
	13	$2.00 \times 10^4$	$1.00 \times 10^5$	$2.00000008 \times 10^4$	$C(4) = 4.99999980 \times 10^{-5}$ farads

As would be expected, solution of the set of thirteen equations proved more difficult than the set of six. In a number of runs, no solutions were achieved. The basic problem appeared to be the failure of the Newton-Raphson method to converge with reasonable values of  $V_{\text{limit}}$  (a value of 100 was used).

#### 4.2.3 Clustering of Sets of Roots

It is of interest to note that in both Table 4-1 and Table 4-2 each individual root within one of the sets of roots is generally of the same order of magnitude as the corresponding root in the other set of roots. In the course of the study it was observed that whenever one set of roots was known, finding a second set was more easily accomplished. There appears to be some tendency for the sets of roots to equations of the type under investigation to "cluster" together in a particular region in the  $p$ -dimensional space defined by the  $p$  unknowns. Thus, after one set of solutions to a particular set of equations has been found, the discovery of a second set near the first appears likely.

## 5.0 CONCLUSIONS AND RECOMMENDATIONS

Based on the research effort to date, the conclusion is reached that, of the numerical methods considered, a combination of the Freudenstein-Roth technique and the Newton-Raphson method represents the most satisfactory method of solving sets of nonlinear algebraic equations characteristic of filter circuit analysis. The digital computer program, which has been developed utilizing this combination, is capable of solving such equations. This program allows considerable leeway in selection of initial estimates of the unknowns, but as the number of unknowns increases, the amount of leeway decreases. For the type of equations under study there is some tendency toward sets of roots clustering together. Thus, obtaining one set of roots to a set of equations tends to facilitate discovery of another set.

The equations which were solved by the program were, as already described, characteristic of filter circuit analysis. The program, however, with only minor modification appears applicable to any set of nonlinear algebraic equations.

Further numerical refinements to the present technique as described in reference 12 appear feasible. Also, a study of the effects of the use of standard or readily available incrementation for all circuit components, to approximate the numerical values of the roots to the equations, is warranted. The recommendation is made that the research effort be continued in accordance with reference 12 in order that the true value of the digital computer program and the numerical procedures contained within it may be established.

APPENDIX A

Source Program

The source program which follows is presented in two versions. The first of these is written in Fortran II and is suitable for running on the IBM 1620, SDS 930, and CDC 3200 digital computers. The second version is written in Fortran IV and is designed for use with the IBM 7094 digital computer. A description of all subroutines is presented in Appendix B. A complete description of all inputs and outputs are provided in Appendix C. A flow diagram of the program is presented in Figure 3-1.

A-1 Fortran II Source Program

```

C      SYSTEM OF N SIMULTANEOUS NONLINEAR EQUATIONS
      DIMENSION IMAX(6),F(6),X(6),C(6),K(6,4,2),FXORIG(6),X1(6),DELX(6),
D      FX(6),DFX(6,6),SUM(6),PSUM(6,6),T(6),P(6,6),A(6,6),R(6),
I      AORIG(6),PHI(6),PTOL(6),FL(6),FC(6),FXI(6),FXLIM(6),
M      XGUES(6),LMAX(6)
100 FORMAT (/4H NA=,I4/)
110 FORMAT(/16H SINGULAR MATRIX/)
120 FORMAT (20I4)
130 FORMAT (2X10HINPUT DATA//21H MAXIMUM NO. OF STEPS,3X,I4/16H NUMBER
1 OF STEPS,8X,I4/20H TIMES THROUGH SIMEQ,4X,I4/15H CONSTANT TERMS/)
140 FORMAT (/32H COMMENCING COEFFICIENT APPROACH)
150 FORMAT (/3X5HFXLIM/(6(4XE16.8)))
180 FORMAT (6H GRID=,I4,3X,4HNOS=,I4)
210 FORMAT (6(4XE16.8))
230 FORMAT (8E10.0)
240 FORMAT (6E12.5)
250 FORMAT ( 3X3H R(,I2,2H)=,E16.8,2X4HOHMS,5X3H L(,I2,2H)=,E16.8,2X
F7HHENRIES,5X3H C(,I2,2H)=,E16.8,2X6HFARADS)
260 FORMAT (38X2HL(,I2,2H)=,E16.8,2X7HHENRIES,6X2HC(,I2,2H)=,E16.8,2X
F6HFARADS)
270 FORMAT (73X3H C(,I2,2H)=,E16.8,2X6HFARADS)
280 FORMAT (37X3H C(,I2,2H)=,E16.8,2X7HHENRIES)
290 FORMAT (3X3H R(,I2,2H)=,E16.8,2X4HOHMS,5X3H L(,I2,2H)=,E16.8,2X7HH
FENRIES)
300 FORMAT (3X3H R(,I2,2H)=,E16.8,2X4HOHMS)
310 FORMAT (3X3H R(,I2,2H)=,E16.8,2X4HOHMS,41X3H C(,I2,2H)=,E16.8,2X6H
FFARADS)
320 FORMAT (/10H VARIABLES/)
330 FORMAT (/72H ALL ROOTS IN THE FOLLOWING SET LIE WITHIN THE PHYSIC
1AL LIMITS SPECIFIED//)
340 FORMAT (49H USING THIS SET OF ESTIMATES, NO ROOTS WERE FOUND//)
350 FORMAT (/75H THE FOLLOWING SET OF ROOTS DO NOT LIE WITHIN THE PHYS
FICAL LIMITS SPECIFIED/)
360 FORMAT (/20H RANGE FOR VARIABLES/3X6HFXORIG/(6(4XE16.8 )))
370 FORMAT (/11H THERE ARE ,I2,15H EQUATIONS AND ,I2,24H UNKNOWNNS,CONS
FISTING OF ,I2,16H RESISTANCE(S), ,I2,19H INDUCTANCE(S),AND ,I2,16H
O CAPACITANCE(S).)
380 FORMAT (85H THE LOWER BOUNDARIES FOR THE RESISTANCES, THE INDUCTAN
FCES, AND THE CAPACITANCES ARE ,2(E16.8,2H, )/5H AND ,E16.8,1H, 48H
O RESPECTIVELY, WHILE THEIR UPPER BOUNDARIES ARE ,2(E16.8,2H, ),4HA
RND /1XE16.8,14H RESPECTIVELY.)
      READ 120, MAXNOS,NOS,KK,JMAX,NMAX,IZMAX,NR,NL,NC,NOR,MR
      READ 120, (IMAX(J),J=1,JMAX)
      READ 120, (LMAX(J),J=1,JMAX)
      READ 240, (F(J),J=1,JMAX)
      READ 230,(PTOL(N),N=1,NMAX)
      READ 230, XRMIN,XLMIN,XCMIN,XRMAX,XLMAX,XCMAX
      READ 230, (FXORIG(N),N=1,NMAX),(FXLIM(N),N=1,NMAX)
      READ 240,(C(M),M=1,NR)
      READ 230, TX
      DO 53 J=1,JMAX
      IJMAX=IMAX(J)
      LJMAX=LMAX(J)
53 READ 120, ((K(J,I,L),L=1,LJMAX),I=1,IJMAX)

```



```

NNOS=NOS
PRINT 130, MAXNOS,NOS, KK
PRINT 210, (F(J), J=1, JMAX)
PRINT 360, FXORIG
PRINT 150, FXLIM
PRINT 370, JMAX, NMAX, NR, NL, NC
PRINT 380, XRMIN, XLMIN, XCMIN, XRMAX, XLMAX, XCMAX
IF (FXORIG(1)-FXLIM(1)) 331, 332, 331
331 CALL ESTIM (NMAX, JMAX, NR, NL, NOR, TX, IMAX, LMAX, F, C, FXORIG, FXLIM, K,
E      FX)
332 PRINT 320
PRINT 210, (FX(N), N=1, NMAX)
NA=1
DO 205 M=1, NMAX
IF (M-NR) 206, 206, 207
207 C(M)=0.
206 NRM=NR+M
205 X(M)=FX(M)-C(M)*FX(NRM)
PRINT 140
51 PRINT 100, NA
DO 48 I=1, NMAX
XGUES(I)=X(I)
48 X1(I)=X(I)
IGRID=1
47 LL=0
ANOS=NOS
KSWTCH=0
LSWTCH=0
NS=0
54 PRINT 180, (IGRID, NOS)
60 DO 3 I=1, NMAX
3 DELX(I)=0.
C  CALCULATE PARTIALS
DO 4 M=1, NMAX
IF (M-NR) 5, 5, 6
6 C(M)=0.
5 NRM=NR+M
FX(M)=C(M)*X(NRM)+X(M)
DO 7 N=1, NMAX
DFX(M, N)=0.
IF (M-N) 9, 8, 9
8 DFX(M, N)=1.
GO TO 7
9 IF (N-(NR+M)) 7, 10, 7
10 DFX(M, N)=C(M)
7 CONTINUE
4 CONTINUE
DO 11 J=1, JMAX
SUM(J)=-F(J)
DO 12 N=1, NMAX
12 PSUM(J, N)=0.
IJMAX=IMAX(J)
DO 13 I=1, IJMAX
T(I)=1.

```

```

      LJMAX=LMAX(J)
      DO 14 L=1,LJMAX
      NK=K(J,I,L)
14    T(I)=T(I)*FX(NK)
      DO 15 N=1,NMAX
      P(I,N)=0.
      DO 16 L=1,LJMAX
      NK=K(J,I,L)
16    P(I,N)=P(I,N)+T(I)*DFX(NK,N)/FX(NK)
C    CALCULATE TOTAL PARTIALS
15    PSUM(J,N)=PSUM(J,N)+P(I,N)
13    SUM(J)=SUM(J)+T(I)
C    DETERMINE LARGEST COEFFICIENT OF EACH EQUATION
      IF(NA-1)17,17,18
17    TX=0.
      IJMAX=IMAX(J)
      DO 19 I=1,IJMAX
      IF(T(I)-TX)19,19,20
20    TX=T(I)
      NX=I
19    CONTINUE
      GO TO 21
18    IF(IMAX(J)+1-NA)22,23,23
22    NX=IMAX(J)
      GO TO 21
23    NX=NA-1
21    IF(KSWTCH-1)24,25,25
C    CALCULATE COEFFICIENTS
24    AORIG(J)=((-SUM(J))/T(NX))+1.
25    IF(LSWTCH-1)30,29,29
30    GRID=IGRID
      IF (AORIG(J)) 1,125,125
      1 A(J,NA)=- (ABS(AORIG(J)-2.))* (1.-GRID/ANOS))+2.
      GO TO 29
125 A(J,NA)=AORIG(J)* (1.-GRID/ANOS)
C    CALCULATE TOTAL PARTIALS (CORRECTED)
29    DO 28 N=1,NMAX
28    PSUM(J,N)=PSUM(J,N)+(A(J,NA)-1.0)*P(NX,N)
11    PHI(J)=- (SUM(J)+(A(J,NA)-1.0)*T(NX))
      KSWTCH=0
      CALL SIMEQ (PSUM,DELX,PHI,JMAX,IE)
      IE=IE+1
      NS=NS+1
      GO TO (31,32),IE
32    PRINT 110
      GO TO 38
31    DO 33 I=1,NMAX
      IF (ABS(DELX(I))-PTOL(I)*ABS(X(I)))33,33,40
33    CONTINUE
      DO 35 I=1,NMAX
35    X(I)=X(I)+DELX(I)
      DO 61 I=1,NMAX
      NRI=NR+I
61    FX(I)=C(I)*X(NRI)+X(I)

```

```

NS=0
LSWTCH=0
KSWTCH=KSWTCH+1
IGRID=IGRID+1
IF (IGRID-NOS-1) 36,52,52
36 DO 26 N=1,NMAX
26 X1(N)=X(N)
GO TO 54
40 KSWTCH=KSWTCH+1
LSWTCH=LSWTCH+1
IF (NS-KK) 37,43,43
37 DO 55 I=1,NMAX
55 X(I)=X(I)+DELX(I)
GO TO 60
43 NOS=2*(NOS+1-IGRID)
IF (NOS-MAXNOS) 44,38,38
44 DO 45 I=1,NMAX
45 X(I)=X1(I)
IGRID=1
GO TO 47
38 NOS=NNOS
PRINT 340
DO 50 I=1,NMAX
50 X(I)=XGUES(I)
211 NA=NA+1
IF (NA-1-IZMAX) 51,49,49
49 MR=0
CALL FCON (MAXNOS,NOS,KK,JMAX,NMAX,NR,LMAX,IMAX,F,PTOL,X,K,C,XGUES
F,FX,IERR)
GO TO (112,52),IERR
52 IJ=0
DO 76 I=1,NR
IF (X(I)-XRMIN) 99,76,76
99 IJ=IJ+1
76 CONTINUE
DO 77 I=1,NR
IF (X(I)-XRMAX) 77,77,101
101 IJ=IJ+1
77 CONTINUE
NRP1=NR+1
NRPNL=NR+NL
DO 102 I=NRP1,NRPNL
IF (X(I)-XLMIN) 103,102,102
103 IJ=IJ+1
102 CONTINUE
DO 104 I=NRP1,NRPNL
IF (X(I)-XLMAX) 104,104,105
105 IJ=IJ+1
104 CONTINUE
DO 106 I=NCC,NMAX
X(I)=1./X(I)
IF (X(I)-XCMIN) 107,106,106
107 IJ=IJ+1
106 CONTINUE

```

```

DO 108 I=NCC,NMAX
IF (X(I)-XCMA) 108,108,109
109 IJ=IJ+1
108 CONTINUE
IF (IJ-1) 111,121,121
111 PRINT 330
GO TO 113
112 PRINT 340
GO TO 67
121 PRINT 350
113 DO 81 I=1,NR
81 R(I)=FX(I)
DO 82 J=NRP1,NRPNL
I=J-NR
82 FL(I)=FX(J)
NCC=NR+NL+1
DO 83 M=NCC,NMAX
I=M-(NR+NL)
FXI(I)=1./FX(M)
83 FC(I)=FXI(I)
IF (NR-NL) 91,72,65
65 IF (NC-NR) 85,74,84
84 MIN=NL
NMIN=NR
MAX=NC
MIN1=MIN+1
NMIN1=NMIN+1
PRINT 250,(I,R(I),I,FL(I),I,FC(I),I=1,MIN)
PRINT 310,(I,R(I),I,FC(I),I=MIN1,NMIN)
PRINT 270,(I,FC(I),I=NMIN1,MAX)
GO TO 67
85 MAX=NR
IF (NC-NL) 88,89,87
87 MIN=NL
NMIN=NC
MIN1=MIN+1
NMIN1=NMIN+1
PRINT 250,(I,R(I),I,FL(I),I,FC(I),I=1,MIN)
PRINT 310,(I,R(I),I,FC(I),I=MIN1,NMIN)
PRINT 300,(I,R(I),I=NMIN1,MAX)
GO TO 67
88 MIN=NC
NMIN=NL
MIN1=MIN+1
NMIN1=NMIN+1
PRINT 250,(I,R(I),I,FL(I),I,FC(I),I=1,MIN)
PRINT 290,(I,R(I),I,FL(I),I=MIN1,NMIN)
PRINT 300,(I,R(I),I=NMIN1,MAX)
GO TO 67
89 NMIN=NC
MIN=NC
NMIN1=NMIN+1
PRINT 250,(I,R(I),I,FL(I),I,FC(I),I=1,MIN)
PRINT 300,(I,R(I),I=NMIN1,MAX)

```

```

GO TO 67
74 MIN=NL
   NMIN=NC
   MIN1=MIN+1
   PRINT 250, (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   PRINT 310, (I,R(I),I,FC(I),I=MIN1,NMIN)
   GO TO 67
91 IF (NL-NC) 96,97,92
92 MAX=NL
   IF (NR-NC) 95,94,93
93 MIN=NC
   NMIN=NR
   MIN1=MIN+1
   NMIN1=NMIN+1
   PRINT 250, (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   PRINT 290, (I,R(I),I,FL(I),I=MIN1,NMIN)
   PRINT 280, (I,FL(I),I=NMIN1,MAX)
   GO TO 67
95 MIN=NR
   NMIN=NC
   MIN1=MIN+1
   NMIN1=NMIN+1
   PRINT 250, (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   PRINT 260, (I,FL(I),I,FC(I),I=MIN1,NMIN)
   PRINT 280, (I,FL(I),I=NMIN1,MAX)
   GO TO 67
94 NMIN=NC
   NMIN1=NMIN+1
   PRINT 250, (I,R(I),I,FL(I),I,FC(I),I=1,NMIN)
   PRINT 280, (I,FL(I),I=NMIN1,MAX)
   GO TO 67
96 MIN=NR
   NMIN=NL
   MAX=NC
   MIN1=MIN+1
   NMIN1=NMIN+1
   PRINT 250, (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   PRINT 260, (I,FL(I),I,FC(I),I=MIN1,NMIN)
   PRINT 270, (I,FC(I),I=NMIN1,MAX)
   GO TO 67
97 MIN=NR
   NMIN=NL
   MIN1=MIN+1
   PRINT 250, (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   PRINT 260, (I,FL(I),I,FC(I),I=MIN1,NMIN)
   GO TO 67
72 IF (NC-NL) 66,90,73
73 MIN=NL
   MAX=NC
   NMIN=NR
   NMIN1=NMIN+1
   PRINT 250, (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   PRINT 270, (I,FC(I),I=NMIN1,MAX)
   GO TO 67

```

```

66 MIN=NC
   NMIN=NR
   MIN1=MIN+1
   PRINT 250,(I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   PRINT 290,(I,R(I),I,FL(I),I=MIN1,NMIN)
   GO TO 67
90 MIN=NR
   PRINT 250,(I,R(I),I,FL(I),I,FC(I),I=1,MIN)
67 IF (MR) 211,212,211
212 STOP
   END

```

```

SUBROUTINE FCON (MAXNOS,NOS,KK,JMAX,NMAX,NR,LMAX,IMAX,F,PTOL,X,K,
F          C,XGUES,FX,IERR)
  DIMENSION IMAX(6),F(6),FORG(6),X(6),DELX(6),K(6,4,2),C(6),SUM(6),
D          XORIG(6),X1(6),FX(6),DFX(6,6),PSUM(6,6),P(6,6),T(6),
I          PHI(6),PTOL(6),FF(6),XGUES(6),LMAX(6)
110 FORMAT(/16H SINGULAR MATRIX/)
180 FORMAT (6H GRID=,I4,3X.4HNOS=,I4)
320 FORMAT (//29H COMMENCING CONSTANT APPROACH//)
   PRINT 320
   IERR=1
   DO 1 I=1,NMAX
     X(I)=XGUES(I)
1   X1(I)=X(I)
   IGRID=1
33 KSWTCH=0
   LSWTCH=0
   ANOS=NOS
   NS=0
22 PRINT 180,IGRID,NOS
43 DO 2 I=1,NMAX
2  DELX(I)=0.
   CALCULATE PARTIALS
   DO 3 M=1,NMAX
     IF(M-NR)4,4,5
5  C(M)=0.
4  NRM=NR+M
   FX(M)=C(M)*X(NRM)+X(M)
   DO 6 N=1,NMAX
     DFX(M,N)=0.
     IF(M-N)8,7,8
7  DFX(M,N)=1.
   GO TO 6
8  IF(N-(NR+M))6,9,6
9  DFX(M,N)=C(M)
6  CONTINUE

```

```

3  CONTINUE
   DO 10 J=1,JMAX
   SUM(J)=0.
   DO 11 N=1,NMAX
11  PSUM(J,N)=0.
   IJMAX=IMAX(J)
   DO 12 I=1,IJMAX
   T(I)=1.
   LJMAX=LMAX(J)
   DO 13 L=1,LJMAX
   NK=K(J,I,L)
13  T(I)=T(I)*FX(NK)
   DO 14 N=1,NMAX
   P(I,N)=0.
   DO 15 L=1,LJMAX
   NK=K(J,I,L)
15  P(I,N)=P(I,N)+T(I)*DFX(NK,N)/FX(NK)
C   CALCULATE TOTAL PARTIALS
14  PSUM(J,N)=PSUM(J,N)+P(I,N)
12  SUM(J)=SUM(J)+T(I)
   IF(KSWTCH-1)28,29,29
C   CALCULATE CONSTANT TERM
28  FORG(J)=SUM(J)
29  IF(LSWTCH-1)40,41,41
40  GRID=IGRID
   IF (FORG(J)) 50,51,51
50  FF(J)=F(J)**(GRID/ANOS)*(-(ABS(FORG(J))+2.*F(J))**(1.-GRID/ANOS))
   1+2.*F(J)
   GO TO 41
51  FF(J)=F(J)**(GRID/ANOS)*FORG(J)**(1.-GRID/ANOS)
41  PHI(J)=-SUM(J)+FF(J)
10  CONTINUE
   KSWTCH=0
   CALL SIMEQ (PSUM,DELX,PHI,JMAX,IE)
   IE=IE+1
   NS=NS+1
   GO TO (16,17),IE
17  PRINT 110
   GO TO 23
16  DO 18 I=1,NMAX
   IF(ABS(DELX(I))-PTOL(I)*ABS(X(I)))18,18,19
18  CONTINUE
   DO 20 I=1,NMAX
20  X(I)=X(I)+DELX(I)
   DO 34 I=1,NMAX
   NRI=NR+I
34  FX(I)=C(I)*X(NRI)+X(I)
   NS=0
   KSWTCH=KSWTCH+1
   LSWTCH=0
   IGRID=IGRID+1
   IF (IGRID-NOS-1) 42,99,99
99  IERR=IERR+1
   RETURN

```

```

42 DO 30 I=1,NMAX
30 X1(I)=X(I)
GO TO 22
19 KSWTCH=KSWTCH+1
LSWTCH=LSWTCH+1
IF(NS-KK)24,25,25
24 DO 26 I=1,NMAX
26 X(I)=X(I)+DELX(I)
GO TO 43
25 NOS=2*(NOS+1-IGRID)
IF(NOS-MAXNOS)31,23,23
31 DO 32 I=1,NMAX
32 X(I)=X1(I)
IGRID=1
GO TO 33
23 DO 35 N=1,NMAX
NRM=NR+N
35 FX(N)=C(N)*X(NRM)+X(N)
36 RETURN
END

```

```

SUBROUTINE ESTIM (NMAX,JMAX,NR,NL,NOR,TX,IMAX,LMAX,F,C,FXORIG,
E FXLIM,JK,FX)
DIMENSION FXORIG(6),FXLIM(6),X(6),X1(6),XP(6),C(6),FX(6),SUM(6),
D F(6),IMAX(6),LMAX(6),JK(6,4,2),T(6),PHI(6),CHX(6),TOL(6)
110 FORMAT (/3X3HLX=,I4)
NCC=NR+NL+1
DO 29 L=NCC,NMAX
FXLIM(L)=1./FXLIM(L)
29 FXORIG(L)=1./FXORIG(L)
JJ=1
LX=0
20 IF (JJ-1) 16,15,16
15 DO 3 J=2,NMAX
3 FX(J)=EXP((ALOG(FXLIM(J)*FXORIG(J)))/2.0)
16 DO 1 K=1,NMAX
DO 2 I=1,NOR
AP=I-1
XNOS=NOR
FX(K)=FXORIG(K)*EXP(AP*ALOG(FXLIM(K)/FXORIG(K)))/(XNOS-1.)
DO 8 M=1,JMAX
SUM(M)=-F(M)
IMMAX=IMAX(M)
DO 9 J=1,IMMAX
T(J)=1.
LMMAX=LMAX(M)
DO 10 N=1,LMMAX
NK=JK(M,J,N)

```



```

10 T(J)=T(J)*FX(NK)
9 SUM(M)=SUM(M)+T(J)
8 PHI(M)=-SUM(M)
  APHI=0.
  DO 11 N=1,JMAX
11 APHI=APHI+ABS(PHI(N))
  IF (I-1) 22,12,22
22 IF (APHI-APHI1) 12,12,13
12 APHI1=APHI
  DO 19 N=1,NMAX
19 X1(N)=FX(N)
2 CONTINUE
  GO TO 26
13 DO 28 N=1,NMAX
28 FX(N)=X1(N)
26 IF (JJ-1) 18,17,18
18 CHX(K)=ABS(ALOG(XP(K)/X1(K)))
  TOL(K)=TX*ALOG(FXLIM(K)/FXORIG(K))/7(XNOS-1.)
  IF (CHX(K)-TOL(K)) 21,21,23
21 LX=LX+1
  IF (LX-NMAX) 30,24,24
23 LX=0
30 PRINT 110, LX
17 DO 14 N=1,NMAX
14 XP(N)=FX(N)
1 CONTINUE
  JJ=JJ+1
  GO TO 20
24 PRINT 110, LX
  DO 25 I=1,NMAX
25 FX(I)=XP(I)
  RETURN
  END

```

```

SUBROUTINE SIMEQ (A,X,B,N,IERR)
C SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS
  DIMENSION A(6,6),X(6),B(6),IND(6)
  DO 1 I=1,N
1 IND(I)=I
  DO 15 K=1,N
C SEARCH ARRAY FOR LARGEST VALUE
  IX=K
  JX=K
  DO 3 I=K,N
  DO 3 J=K,N
  IF(ABS(A(I,J))-ABS(A(IX,JX))) 3,3,2
2 IX=I
  JX=J
3 CONTINUE

```

```

      IF (A(IX,JX)) 5,4,5
4  IERR=1
   RETURN
5  IF (IX-K) 8,8,6
   EXCHANGE ROWS
6  DO 7 J=K,N
   TEMP=A(IX,J)
   A(IX,J)=A(K,J)
7  A(K,J)=TEMP
   TEMP=B(IX)
   B(IX)=B(K)
   B(K)=TEMP
8  IF (JX-K) 11,11,9
   EXCHANGE COLUMNS
9  DO 10 I= 1,N
   TEMP=A(I,JX)
   A(I,JX)=A(I,K)
10 A(I,K)=TEMP
   INDEX=IND(JX)
   IND(JX)=IND(K)
   IND(K)=INDEX
11 PIVOT=A(K,K)
   DO 12 J=K,N
12 A(K,J)=A(K,J)/PIVOT
   B(K)=B(K)/PIVOT
   DO 15 I=1,N
   IF (I-K) 13,15,13
13 TEMP=A(I,K)
   DO 14 J=K,N
14 A(I,J)=A(I,J)-A(K,J)*TEMP
   B(I)=B(I)-B(K)*TEMP
15 CONTINUE
   DO 16 I=1,N
   INDEX=IND(I)
16 X(INDEX)=B(I)
   IERR=0
   RETURN
END

```

A-2 Fortran IV Source Program

```

    DIMENSION IMAX(6),F(6),X(6),C(6),K(6,4,2),FXORIG(6),X1(6),DELX(6),
    D      FX(6),DFX(6,6),SUM(6),PSUM(6,6),T(6),P(6,6),A(6,6),R(6),
    I      AORIG(6),PHI(6),PTOL(6),FL(6),FC(6),FXI(6),FXLIM(6),
    M      XJUES(6),LMAX(6)
100 F0RMA7 (/4H NA=,I4/)
110 F0RMA7 (/16H SINGULAR MATRIX/)
120 F0RMA7 (20I4)
130 F0RMA7 (2X10H INPUT DATA//21H MAXIMUM NO. 0F STEPS,3X,I4/16H NUMBER
    1 0F STEPS,8X,I4/20H TIMES THRU0GH SIMEQ,4X,I4/15H C0NSTANT TERMS/)
140 F0RMA7 (/32H C0MMENCING C0EFFICIENT APPR0ACH)
150 F0RMA7 (/3X5HFXLIM/(6(5XF12.2)))
180 F0RMA7 (6H GRID=,I4,3X,4HN0S=,I4)
210 F0RMA7 (6(4XE16.8))
230 F0RMA7 (8E10..)
240 F0RMA7 (6E12.5)
250 F0RMA7 ( 3X3H R(,I2,2H)=,E16.8,2X4H0HMS,5X3H L(,I2,2H)=,E16.8,2X
    F7HHENRIES,5X3H C(,I2,2H=,E16.8,2X6HFARADS)
260 F0RMA7 (38X2HL(,I2,2H)=,E16.8,2X7HHENRIES,6X2HC(,I2,2H)=,E16.8,2X
    F0HFARADS)
270 F0RMA7 (73X3H C(,I2,2H)=,E16.8,2X6HFARADS)
280 F0RMA7 (37X3H C(,I2,2H)=,E16.8,2X7HHENRIES)
290 F0RMA7 (3X3H R(,I2,2H)=,E16.8,2X4H0HMS,5X3H L(,I2,2H)=,E16.8,2X7HH
    FENRIES)
300 F0RMA7 (3X3H R(,I2,2H)=,E16.8,2X4H0HMS)
310 F0RMA7 (3X3H R(,I2,2H)=,E16.8,2X4H0HMS,41X3H C(,I2,2H)=,E16.8,2X6H
    FFARADS)
320 F0RMA7 (/10H VARIABLES/)
330 F0RMA7 (/72H ALL R00TS IN THE F0LL0WING SET LIE W1THIN THE PHYSIC
    IAL LIMITS SPECIFIED/)
340 F0RMA7 (49H USING THIS SET 0F ESTIMATES, N0 R00TS WERE F0UND/)
350 F0RMA7 (/75H THE F0LL0WING SET 0F R00TS D0 N0T LIE WITHIN THE PHYS
    ICAL LIMITS SPECIFIED/)
360 F0RMA7 (/20H RANGE F0R VARIABLES/3X6HFXORIG/(6(5X,F12.2)))
370 F0RMA7 (/11H THERE ARE ,I2,15H EQUATIONS AND ,I2,24H UNKN0WNS,C0NS
    FISTING 7F ,I2,14H RESISTANCE(S), ,I2,19H INDUCTANCE(S),AND ,I2,16H
    0 CAPACITANCE(S).)
380 F0RMA7 (85H THE L0WER B0UNDARIES F0R THE RESISTANCES, THE INDUCTAN
    CES, AND THE CAPACITANCES ARE ,2(E16.8,2H, )/5H AND ,E16.8,1H, 48H
    0 RESPECTIVELY, WHILE THEIR UPPER B0UNDARIES ARE ,2(E16.8,2H, ),4HA
    RND /1XE16.8,14H RESPECTIVELY.)
    READ (5,120) MAXN0S,N0S,KK,JMAX,NMAX,IZMAX,NR,NL,N0,N0R,MR
    READ (5,120) (IMAX(J),J=1,JMAX)
    READ (5,120) (LMAX(J),J=1,JMAX)
    READ (5,240) (F(J),J=1,JMAX)
    READ (5,230) (PTOL(N),N=1,NMAX)
    READ (5,230) XRMIN,XLMIN,XCMIN,XRMAX,XLMAX,XCMAX
    READ (5,230) (FXORIG(N),N=1,NMAX),(FXLIM(N),N=1,NMAX)
    READ(5,240) (C(M),M=1,NR)
    READ (5,230) IX
    D0 53 J=1,JMAX
    IJMAX=IMAX(J)
    LJMAX=LMAX(J)
53 READ (5,120) ((K(J,I,L),L=1,LJMAX),I=1,IJMAX)
    NN0S=N0S
    WRITE (5,130) MAXN0S,N0S,KK

```

```

WRITE (6,210) (F(J),J=1,JMAX)
WRITE (6,360) FXORIG
WRITE (6,150) FXLIM
WRITE (6,370) JMAX,NMAX,NR,NL,NC
WRITE(6,380)XRMIN,XLMIN,XCMIN,XRMAX,XLMAX,XCMAX
IF (FXORIG(1)-FXLIM(1)) 331,332,331
331 CALL ESTIM (NMAX,JMAX,NR,NL,NØR,IX,IMAX,LMAX,F,C,FYØRIG,FXLIM,K,
E      FX)
332 WRITE (6,320)
WRITE (6,210) (FX(N),N=1,NMAX)
NA=1
DØ 205 M=1,NMAX
IF (M-NR) 206,206,207
207 C(M)=0.
206 NRM=NR+M
205 X(M)=FX(M)-C(M)*FX(NRM)
WRITE (6,140)
51 WRITE (6,100) NA
DØ 48 I=1,NMAX
XGUES(I)=X(I)
48 X1(I)=X(I)
IGRID=1
47 LL=0
ANØS=NØS
KSWICH=0
LSWICH=0
NS=0
54 WRITE (6,180) IGRID,NØS
60 DØ 3 I=1,NMAX
3 DELX(I)=0.
C CALCULATE PARTIALS
DØ 4 M=1,NMAX
IF(M-NR)5,5,6
6 C(M)=0.
5 NRM=NR+M
FX(M)=C(M)*X(NRM)+X(M)
DØ 7 N=1,NMAX
DFX(M,N)=0.
IF(M-N)0,8,9
8 DFX(M,N)=1.
6Ø TØ 7
9 IF(N-(NR+M))7,10,7
10 DFX(M,N)=C(M)
7 CØNTINUE
4 CØNTINUE
DØ 11 J=1,JMAX
SUM(J)=-F(J)
DØ 12 N=1,NMAX
12 PSUM(J,N)=0.
IJMAX=IMAX(J)
DØ 13 I=1,IJMAX
T(I)=1.
LJMAX=LMAX(J)
DØ 14 L=1,LJMAX
NK=K(J,I,L)
14 T(I)=T(I)*FX(NK)

```

```

      DO 15 N=1,NMAX
      P(I,N)=0.
      DO 16 L=1,LJMAX
      NK=K(J,I,L)
16  P(I,N)=P(I,N)+T(I)*DFX(NK,N)/FX(NK)
C   CALCULATE TOTAL PARTIALS
15  PSUM(J,N)=PSUM(J,N)+P(I,N)
13  SUM(J)=SUM(J)+T(I)
C   DETERMINE LARGEST COEFFICIENT OF EACH EQUATION
      IF(NA-1)17,17,18
17  TX=0.
      IJMAX=IMAX(J)
      DO 19 I=1,IJMAX
      IF(T(I)-TX)19,19,20
20  TX=T(I)
      NX=I
19  CONTINUE
      GO TO 21
18  IF(IMAX(J)+1-NA)22,23,23
22  NX=IMAX(J)
      GO TO 21
23  NX=NA-1
21  IF(KSWTCH-1)24,25,25
C   CALCULATE COEFFICIENTS
24  AORIG(J)=((-SUM(J))/T(NX))+1.
25  IF(LSWTCH-1)30,29,29
30  GRID=IGRID
      IF (AORIG(J)) 1,125,125
      1 A(J,NA)=-(ABS(AORIG(J)-2.)*(1.-GRID/ANOS))+2.
      GO TO 29
125 A(J,NA)=AORIG(J)*(1.-GRID/ANOS)
C   CALCULATE TOTAL PARTIALS (CORRECTED)
29  DO 28 N=1,NMAX
28  PSUM(J,N)=PSUM(J,N)+(A(J,NA)-1.0)*P(NX,N)
11  PHI(J)=- (SUM(J)+(A(J,NA)-1.0)*T(NX))
      KSWTCH=0
      CALL SIMPQ (PSUM,DELX,PHI,JMAX,IE)
      IE=IE+1
      NS=NS+1
      GO TO (31,32),IE
32  WRITE (5,110)
      GO TO 30
31  DO 33 I=1,NMAX
      IF (ABS(DELX(I))-PTOL(I)*ABS(X(I)))33,33,40
33  CONTINUE
      DO 35 I=1,NMAX
35  X(I)=X(I)+DELX(I)
      DO 61 I=1,NMAX
      NRI=NR+1
61  FX(I)=C(I)*X(NRI)+X(I)
      NS=1
      LSWTCH=0
      KSWTCH=KSWTCH+1
      IGRID=IGRID+1
      IF(IGRID-NOS-1)36,52,52
36  DO 26 N=1,NMAX

```

```

26 X1(N)=X(N)
   GO TO 34
40 KSWTCH=KSWTCH+1
   LSWTCH=LSWTCH+1
   IF (NS-KK) 37,43,43
37 DO 55 I=1,NMAX
55 X(I)=X(I)+DELX(I)
   GO TO 6
43 N0S=2*(N0S+1-IGRID)
   IF(N0S-MAXN0S)44,38,38
44 DO 45 I=1,NMAX
45 X(I)=X1(I)
   IGRID=1
   GO TO 47
38 N0S=NN0S
   WRITE (6,340)
   DO 50 I=1,NMAX
50 X(I)=XGUES(I)
211 NA=NA+1
   IF(JA-1-IZMAX)51,49,49
49 MR=0
   CALL FC0N (MAXN0S,N0S,KK,JMAX,NMAX,NR,LMAX,IMAX,F,PT0L,X,K,C,XGUES
   F
   ,FX,IERR)
   GO TO (112,52),IERR
52 IJ=0
   DO 76 I=1,NR
   IF (X(I)-XRMIN) 99,76,76
99 IJ=IJ+1
76 CONTINUE
   DO 77 I=1,NR
   IF (X(I)-XRMAX) 77,77,101
101 IJ=IJ+1
77 CONTINUE
   NRPI=NR+1
   NRPNL=NR+NL
   DO 102 I=NRPI,NRPNL
   IF (X(I)-XLMIN) 103,102,102
103 IJ=IJ+1
102 CONTINUE
   DO 104 I=NRPI,NRPNL
   IF(X(I)-XLMAX) 104,104,105
105 IJ=IJ+1
104 CONTINUE
   DO 106 I=NCC,NMAX
   X(I)=1./X(I)
   IF (X(I)-XCMIN) 107,106,106
107 IJ=IJ+1
106 CONTINUE
   DO 108 I=NCC,NMAX
   IF (X(I)-XC MAX) 108,108,109
109 IJ=IJ+1
108 CONTINUE
   IF (IJ-1) 111,121,121
111 WRITE (6,330)
   GO TO 113
112 WRITE (6,340)

```

```

121 WRITE (6,350)
    GO TO 67
113 DO 81 I=1,NR
81  R(I)=FX(I)
    DO 82 J=NRP1,NRPNL
    I=J-NR
82  FL(I)=FX(J)
    NCC=NR+NL+1
    DO 83 M=NCC,NMAX
    I=M-(NR+NL)
    FXI(I)=1./FX(M)
83  FC(I)=FXI(I)
    IF (NR-NL) 91,72,65
65  IF (NC-NR) 85,74,84
84  MIN=NL
    NMIN=NR
    MAX=NC
    MIN1=MIN+1
    NMIN1=NMIN+1
    WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
    WRITE (6,310) (I,R(I),I,FC(I),I=MIN1,NMIN)
    WRITE (6,270) (I,FC(I),I=NMIN1,MAX)
    GO TO 67
85  MAX=NR
    IF (NC-NL) 88,89,87
87  MIN=NL
    NMIN=NC
    MIN1=MIN+1
    NMIN1=NMIN+1
    WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
    WRITE (6,310) (I,R(I),I,FC(I),I=MIN1,NMIN)
    WRITE (6,300) (I,R(I),I=NMIN1,MAX)
    GO TO 67
88  MIN=NC
    NMIN=NL
    MIN1=MIN+1
    NMIN1=NMIN+1
    WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
    WRITE (6,290) (I,R(I),I,FL(I),I=MIN1,NMIN)
    WRITE (6,300) (I,R(I),I=NMIN1,MAX)
    GO TO 67
89  NMIN=NC
    MIN=NC
    NMIN1=NMIN+1
    WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
    WRITE (6,300) (I,R(I),I=NMIN1,MAX)
    GO TO 67
74  MIN=NL
    NMIN=NC
    MIN1=MIN+1
    WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
    WRITE (6,310) (I,R(I),I,FC(I),I=MIN1,NMIN)
    GO TO 67
91  IF (NL-NC) 96,97,92
92  MAX=NL
    IF (NR-NC) 95,94,93

```



```

93 MIN=NC
   NMIN=NR
   MIN1=MI+1
   NMIN1=NMIN+1
   WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   WRITE (6,290) (I,R(I),I,FL(I),I=MIN1,NMIN)
   WRITE (6,280) (I,FL(I),I=NMIN1,MAX)
   GO TO 67

95 MIN=NR
   NMIN=NC
   MIN1=MI+1
   NMIN1=NMIN+1
   WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   WRITE (6,260) (I,FL(I),I,FC(I),I=MIN1,NMIN)
   WRITE (6,280) (I,FL(I),I=NMIN1,MAX)
   GO TO 67

94 NMIN=NC
   NMIN1=NMIN+1
   WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,NMIN)
   WRITE (6,280) (I,FL(I),I=NMIN1,MAX)
   GO TO 67

96 MIN=NR
   NMIN=NL
   MAX=NC
   MIN1=MIN+1
   NMIN1=NMIN+1
   WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   WRITE (6,260) (I,FL(I),I,FC(I),I=MIN1,NMIN)
   WRITE (6,270) (I,FC(I),I=NMIN1,MAX)
   GO TO 67

97 MIN=NR
   NMIN=NL
   MIN1=MIN+1
   WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   WRITE (6,260) (I,FL(I),I,FC(I),I=MIN1,NMIN)
   GO TO 67

72 IF (NC-NL) 66,90,73
73 MIN=NL
   MAX=NC
   NMIN=NR
   NMIN1=NMIN+1
   WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   WRITE (6,270) (I,FC(I),I=NMIN1,MAX)
   GO TO 67

66 MIN=NC
   NMIN=NR
   MIN1=MIN+1
   WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
   WRITE (6,290) (I,R(I),I,FL(I),I=MIN1,NMIN)
   GO TO 67

90 MIN=NR
   WRITE (6,250) (I,R(I),I,FL(I),I,FC(I),I=1,MIN)
67 IF (MR) 211,212,211
212 STOP
   END

```

```

SUBROUTINE SIMEQ (A,X,B,N,IERR)
C  SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS
  DIMENSION A(6,6),X(6),B(6),IND(6)
  DO 1 I=1,N
1  IND(I)=I
  DO 15 K=1,N
C  SEARCH ARRAY FOR LARGEST VALUE
    IX=K
    JX=K
    DO 3 I=K,N
      DO 3 J=K,N
        IF (ABS(A(I,J))-ABS(A(IX,JX))) 3,3,2
2      IX=I
        JX=J
3    CONTINUE
    IF (A(IX,JX)) 5,4,5
4    IERR=1
    RETURN
5    IF (IX-K) 8,8,6
C  EXCHANGE ROWS
6    DO 7 J=K,N
      TEMP=A(IX,J)
      A(IX,J)=A(K,J)
7    A(K,J)=TEMP
      TEMP=B(IX)
      B(IX)=B(K)
      B(K)=TEMP
8    IF (JX-K) 11,11,9
C  EXCHANGE COLUMNS
9    DO 10 I= 1,N
      TEMP=A(I,JX)
      A(I,JX)=A(I,K)
10   A(I,K)=TEMP
      INDEX=IND(JX)
      IND(JX)=IND(K)
      IND(K)=INDEX
11   PIVOT=A(K,K)
      DO 12 J=K,N
12   A(K,J)=A(K,J)/PIVOT
      B(K)=B(K)/PIVOT
      DO 15 I=1,N
        IF (I-K) 13,15,13
13   TEMP=A(I,K)
      DO 14 J=K,N
14   A(I,J)=A(I,J)-A(K,J)*TEMP
      B(I)=B(I)-B(K)*TEMP
15   CONTINUE
      DO 16 I=1,N
        INDEX=IND(I)
16   X(INDEX)=B(I)
      IERR=0
      RETURN
      END

```

```

SUBROUTINE FC0N (MAXN0S,N0S,KK,JMAX,NMAX,NR,LMAX,IMAX,F,PT0L,X,K,
I          C,XGUES,FX,IERR)
DIMENSION IMAX(6),F(6),F0RG (6),X(6),DELX(6),K(6,4,2),C(6),SUM(6),
D          X0RIG(6),X1(6),FX(6),DFX(6,6),PSUM(6,6),P(6,6),T(6),
I          PHI(6),PT0L(6),FF(6),XGUES(6),LMAX(6)
110 F0RMAT (/16H SINGULAR MATRIX/)
180 F0RMAT (16H GRID=,I4,3X,4HN0S=,I4)
320 F0RMAT (/20H COMMENCING CONSTANT APPROACH/)
WRITE (6,320)
IERR=1
D0 1 I=1,NMAX
X(I)=XGUES(I)
1 X1(I)=X(I)
IGRID=1
33 KSWTCH=0
LSWTCH=0
AN0S=N0S
NS=0
22 WRITE (6,180) IGRID,N0S
43 D0 2 I=1,NMAX
2 DELX(I)=0.
C CALCULATE PARTIALS
D0 3 M=1,NMAX
IF(M-NR)4,4,5
5 C(M)=0.
4 NRM=NR+M
FX(M)=C(M)*X(NRM)+X(M)
D0 6 N=1,NMAX
DFX(M,N)=0.
IF(M-N)8,7,6
7 DFX(M,N)=1.
G0 T0 6
8 IF(N-(NR+M))6,9,6
9 DFX(M,N)=C(M)
6 C0NTINUE
3 C0NTINUE
D0 10 J=1,JMAX
SUM(J)=0.
D0 11 N=1,NMAX
11 PSUM(J,N)=0.
IJMAX=IMAX(J)
D0 12 I=1,IJMAX
T(I)=1.
LJMAX=LMAX(J)
D0 13 L=1,LJMAX
NK=K(J,I,L)
13 T(I)=T(I)*FX(NK)
D0 14 N=1,NMAX
P(I,N)=0.
D0 15 L=1,LJMAX
NK=K(J,I,L)
15 P(I,N)=P(I,N)+T(I)*DFX(NK,N)/FX(NK)
C CALCULATE TOTAL PARTIALS
14 PSUM(J,N)=PSUM(J,N)+P(I,N)
12 SUM(J)=SUM(J)+T(I)

```

```

      IF(KSWTCH-1)28,29,29
      CALCULATE CONSTANT TERM
28  FØRG (J)=SUM(J)
29  IF(LSWTCH-1)40,41,41
40  GRID=IGRID
      IF (FØRG (J)) 50,51,51
50  FF(J)=F(J)**(GRID/ANØS)*(-(ABS(FØRG(J))+2.*F(J))**(1.-GRID/ANØS))
51  FF(J)=F(J)**(GRID/ANØS)*FØRG(J)**(1.-GRID/ANØS)
      1+2.*F(J)
      GØ TØ 41
41  PHI(J)=-SUM(J)+FF(J)
10  CØNTINUE
      KSWTCH=0
      CALL SIMFQ (PSUM,DELX,PHI,JMAX,IE)
      IE=IE+1
      NS=NS+1
      GØ TØ (16,17),IE
17  WRITE (6,110)
      GØ TØ 23
16  DØ 18 I=1,NMAX
      IF(ABS(DELX(I))-PTØL(I)*ABS(X(I)))18,18,19
18  CØNTINUE
      DØ 20 I=1,NMAX
20  X(I)=X(I)+DELX(I)
      DØ 34 I=1,NMAX
      NRI=NR+I
34  FX(I)=C(I)*X(NRI)+X(I)
      NS=0
      KSWTCH=KSWTCH+1
      LSWTCH=0
      IGRID=IGRID+1
      IF (IGRID-NØS-1) 42,99,99
99  IERR=IERK+1
      RETURN
42  DØ 30 I=1,NMAX
30  X1(I)=X(I)
      GØ TØ 23
19  KSWTCH=KSWTCH+1
      LSWTCH=LSWTCH+1
      IF(NS-KK)24,25,25
24  DØ 26 I=1,NMAX
26  X(I)=X(I)+DELX(I)
      GØ TØ 43
25  NØS=2*(NØS+1-IGRID)
      IF(NØS-MAXNØS)31,23,23
31  DØ 32 I=1,NMAX
32  X(I)=X1(I)
      IGRID=1
      GØ TØ 35
23  DØ 35 N=1,NMAX
      NRM=NR+1
35  FX(N)=C(N)*X(NRM)+X(N)
36  RETURN
      END

```

```

SUBROUTINE ESTIM (NMAX,JMAX,NR,NL,NØR,IX,IMAX,LMAX,F,C,FXØRIG,
F
FXLIM,JK,FX)
DIMENSION FXØRIG(6),FXLIM(6),X(6),X1(6),XP(6),C(6),FX(6),SUM(6),
D
F(6),IMAX(6),LMAX(6),JK(6,4,2),T(6),PHI(6),CHX(6),TØL(6)
110 FORMAT (/3X3HLX=,I4)
NCC=NR+NL+1
DØ 29 L=NCC,NMAX
FXLIM(L)=1./FXLIM(L)
29 FXØRIG(L)=1./FXØRIG(L)
JJ=1
LX=0
20 IF (JJ-1) 16,15,16
15 DØ 3 J=2,NMAX
3 FX(J)=EXP((ALØG(FXLIM(J)*FXØRIG(J)))/2.0)
16 DØ 1 K=1,NMAX
DØ 2 I=1,NØR
AP=I-1
XNØS=NØR
FX(K)=FXØRIG(K)*EXP(AP*ALØG(FXLIM(K)/FXØRIG(K)))/(XNØS-1.)
DØ 8 M=1,JMAX
SUM(M)=-F(M)
IMMAX=IMAX(M)
DØ 9 J=1,IMMAX
T(J)=1.
LMMAX=LMAX(M)
DØ 10 N=1,LMMAX
NK=JK(M,J,N)
10 T(J)=T(J)*FX(NK)
9 SUM(M)=SUM(M)+T(J)
8 PHI(M)=-SUM(M)
APHI=0.
DØ 11 N=1,JMAX
11 Aphi=APHI+ABS(PHI(N))
IF (I-1) 22,12,22
22 IF (APHI-APHI1) 12,12,13
12 Aphi1=APHI
DØ 19 N=1,NMAX
19 X1(N)=FX(N)
2 CONTINUE
GØ 1Ø 2-
13 DØ 18 N=1,NMAX
28 FX(N)=X1(N)
26 IF (JJ-1) 18,17,18
18 CHX(K)=ABS(ALØG(XP(K)/X1(K)))
TØL(K)=TX*ALØG(FXLIM(K)/FXØRIG(K)))/(XNØS-1.)
IF (CHX(K)-TØL(K)) 21,21,23
21 LX=LX+1
IF (LX-NMAX) 30,24,24
23 LX=0
30 WRITE (6,110) LX
17 DØ 14 N=1,NMAX
14 XP(N)=FX(N)
1 CONTINUE
JJ=JJ+1
GØ TØ 2-

```

```
24 WRITE (6,110) LX  
    DO 25 I=1,NMAX  
25 FX(I)=XP(I)  
    RETURN  
    END
```

## APPENDIX B

Subroutines

In the main program the following subroutines are used:

ESTIM

SIMEQ

FCON

The remainder of this appendix is devoted to descriptions of these subroutines.

B-1 ESTIM (Selection of Initial Estimates)

This subroutine is a technique for obtaining a set of estimates for the variables. The range of interest and the number of increments to be taken for each variable are in the list of arguments of the subroutine ESTIM. (N-1) of the variables are held at a logarithmic mean of the given range, while the Nth variable is varied according to the number of increments. This process is repeated for each variable, with each of the previous variables maintaining their generated value rather than their mean value, as described in Section 2.4.

The N variables are subjected to this process repeatedly until each lies within a desired tolerance of the preceeding corresponding variable. When all N variables satisfy this requirement, they are returned as the desired set of estimates.

Calling Sequence

CALL ESTIM (NMAX, JMAX, NR, NL, NØR, TX, IMAX, LMAX, F, C,FXØRIG,FXLIM, JK,FX)

where

- NMAX - the number of variables
- JMAX - the number of equations in the system
- NR - the number of resistances
- NL - the number of inductances
- NØR - the number of increments to be taken within the specified range
- TX - the tolerance used to determine the acceptability of each variable
- IMAX - maximum number of terms per equation
- IMAX - maximum number of factors per term
- F - the constant values of each equation
- C - the linear relationship between the inductances and their corresponding resistances
- FXORIG - the lower boundary of the range for the variables
- FXLIM - the upper boundary of the range for the variables
- JK - the subscripted subscripts of each variable; these describe the equations within the system
- FX - the outlet for the generated estimates.

#### B-2 SIMEQ (Simultaneous Equation Solver)

This routine employs a Gaussian technique of reducing a coefficient matrix by the pivotal method. The elements themselves are partial differentials resulting from the main program. The largest numerical element is sought and used as the pivoting element. Should this largest element be trivial, an error message is returned, printed out, and the program proceeds as indicated in Figure 3-1.



Calling Sequence

CALL SIMEQ (A, X, B, N, IERR)

where

A - the N x N array in which the elements of the matrix are stored

X - the resulting column matrix

B - the (N+1) column array

N - the number of equations to be solved

IERR - the control indicating singular matrix if IERR=1.

B-3 FCON (Constant Approach)

Like the main program this approach toward solving the system of nonlinear equations employs the Newton-Raphson and Freudenstein-Roth methods. The difference lies in the fact that the constant terms are incremented (or decremented) rather than the leading coefficient of each equation. This method may be used whether or not the coefficient approach fails. The original set of estimates obtained from ESTIM is used, and all the input data are recalled.

Calling Sequence

CALL FCON (MAXNOS, NOS, KK, JMAX, NMAX, NR, LMAX, IMAX, F, PTOL, X, K, C, XGUES, FX, IERR)

where

MAXNOS - maximum number of incrementing (or decrementing) steps to be taken

NOS - the initial number of steps to be taken

**NORTHROP SPACE LABORATORIES**

---

KK     - maximum number of times allowed for subroutine SIMEQ to be called

JMAX   - number of equations

NMAX   - number of variables

NR     - number of resistances

LMAX   - maximum number of factors per term

IMAX   - maximum number of terms per equation

F       - constant terms of each equation

PTØL   - percentage of tolerance allowed on each change of each variable

X       - the variables

K       - subscripted subscripts of each variable which describes the system  
          of equations

C       - linear relationship between inductances and their corresponding  
          resistances

XGUES   - original set of estimates

FX      - outlet for roots, or last set of variables

IERR    - control for determining if roots are found

## APPENDIX C

Description of Computer Inputs & OutputsC-1 Input Nomenclature

All inputs are made through the familiar FORTRAN commands. The inputs for the various subroutines are read into the main program. The following is an alphabetic listing and a description of each input item used. The symbol in brackets following each description refers to the nomenclature used in Section 2.0.

C(M)	Constant for establishing linear relationship between resistances and inductances, ( $1 \leq M \leq NR$ ). $[K_m]$
F(J)	The constant terms of each equation, ( $1 \leq J \leq JMAX$ ). $[F_j]$
FXLIM(N)	Upper limit on range of variables used to obtain estimates, ( $1 \leq N \leq NMAX$ ). $[Y_{n(up)}]$
FXØRIG(N)	Lower limit on range of variables used to obtain estimates, ( $1 \leq N \leq NMAX$ ). $[Y_{n(low)}]$
IMAX(J)	Number of terms in each equation, ( $1 \leq J \leq JMAX$ ). $[Q_j]$
IZMAX	Number of column arrangements to be used in the coefficient approach. $[Q_{limit}]$
JMAX	Number of equations to be solved. $[p]$
K(J,I,L)	Subscript for each factor $[1 \leq L \leq LMAX(J)]$ of each term $[1 \leq I \leq IMAX(J)]$ of each equation ( $1 \leq J \leq JMAX$ ). $[r(j,i,k)]$

KK                    Maximum number of times subroutine SIMEQ may be called for each attempt at convergence.  $[U_{\text{limit}}]$

LMAX(J)             Number of factors per term per equation,  $(1 \leq J \leq \text{JMAX})$ .  $[d_j]$

MAXNØS             The maximum number of increments allowed in the Freudenstein-Roth Technique.  $[V_{\text{limit}}]$

MR                   Input option relating to search for satisfactory roots. If MR=0, the computer will stop after one set of satisfactory roots have been obtained. If MR=1, the computer will continue searching for roots until IZMAX is reached.

NC                   Number of variables representing capacitances.  $[w]$

NL                   Number of variables representing inductances.  $[v]$

NMAX                Number of variables involved.  $[p]$

NOR                  Number of increments to be taken from FXORIG(N) to FXLIM(N) in selection of initial estimate.  $[W]$

NOS                  Number of increments to be used in varying the coefficients or constants for each equation in the Freudenstein-Roth technique.  $[V]$

NR                   Number of variables representing resistances.  $[u]$

PTØL(N)            Maximum percentage of change in variables for establishing convergence in Newton-Raphson method  $(1 \leq N \leq \text{NMAX})$ .

TX	Constant for determining convergence in selection of initial estimate of each variable. $[\Delta a]$
XCMAX	Maximum value variables representing capacitances can attain
XCMIN	Minimum value variables representing capacitances can attain
XLMAX	Maximum value variables representing inductances can attain
XLMIN	Minimum value the variables representing inductances can attain
XRMAX	Maximum value variables representing resistances can attain
XRMIN	Minimum value the variables representing resistances can attain

#### C-2 Input Format

The correct format for the inputs already described is provided on the Fortran key punch forms which follow. Representative values of the inputs involved are listed immediately below the Fortran symbol. The values correspond to the set of six equations and six unknowns presented in Appendix D.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
84

72

# KEY PUNCH FORM - GENERAL PURPOSE

FD-400 (Rev. 10-1-59)

SIMULTANEOUS SOLUTION OF SETS OF NONLINEAR ALGEBRAIC EQUATIONS										ENGINEER	2	2
ACCOUNT NO.	SALES ORDER	DASH	FILE NO.	FOR CHGN. NO.						ANALYST		2/28/66
<div> <div>C(1)</div> <div>C(2)</div> <div>C(3)</div> </div>												
<div> <div>1.0</div> <div>1.0</div> <div>0.0</div> </div>												
TX												
<div>1.0</div>												
K(111)K(112)												
4 5												
K(211)K(212) K(221) K(222) K(231) K(232) K(241) K(242)												
2 4 3 4 1 5 3 5												
K(311) K(312) K(321) K(322) K(331) K(332) K(341) K(342)												
5 6 1 2 1 3 2 3												
K(411) K(412) K(421) K(422)												
3 6 2 6												
K(511) K(512)												
3 5												
K(611) K(612)												
2 3												

C-3 Output Nomenclature

The first data printed out are certain original input data. The "Maximum No. of Steps" referred to is MAXNOS; the "Number of Steps" is NOS; and the "Times Through SIMEQ" is KK. The "Constant Terms" are F(J) arranged in order of subscripts reading from left to right. Following these terms the range of interest for each variable is established by means of FXORIG(J) and FXLIM(J) which are arranged in the same manner as F(J). The rest of the printout of input data describes the number of equations and unknowns, the number of resistances, inductances, and capacitances involved, and the minimum and maximum values of such components.

After the printout of input data, the printout is designed to indicate to the computer operator the steps taken by the computer in obtaining a solution. The terminology used is the same as that already provided in the description of input nomenclature with the following additions:

- |      |                                                                                                                                                                                                                                                                                      |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GRID | The iterative step number in the Freudenstein-Roth technique<br>( $1 \leq \text{GRID} \leq \text{NOS}$ )                                                                                                                                                                             |
| LX   | The counter used in the process of selecting initial estimates.<br>When $\text{LX} = \text{NMAX}$ the selection process is complete.                                                                                                                                                 |
| NA   | The counter used in the coefficient approach to indicate the arrangement of terms in the set of equations. When $\text{NA}=1$ , the largest term in each equation is used as the first term. When $\text{NA}=2$ , the original order of terms as input is used. When $\text{NA}=3$ , |



the original first term in each equation is exchanged with the second term. When  $NA=4$ , the original first term is exchanged with the third term, etc.

The final output depends upon conditions arising within the program. Should a satisfactory set of roots (a set in which all elements lie within the physical limits specified) be obtained, a statement indicating this fact is printed out together with the roots appropriately denoted as resistances, capacitances or inductances. In the case in which roots are found but are not acceptable, a statement indicates this fact followed by a listing of the values of the unsatisfactory roots. If no satisfactory roots are found by the time all IZMAX columns have been tried, a statement is printed to this effect. As already noted the program contains an option that, in case a set of roots are found, the process either stops or continues searching using the remaining of the IZMAX columns. If a singular matrix is encountered in the SIMEQ subroutine, the words "SINGULAR MATRIX" are printed out, and the computer proceeds as it would if no roots had been found, as indicated in Figure 3-1.

#### C-4 Sample Output

The output which follows is based on the input data previously presented in this appendix for six equations and six unknowns.

# INPUT DATA

MAXIMUM NO. OF STEPS 100  
NUMBER OF STEPS 25  
TIMES THROUGH SIMEQ 20  
CONSTANT TERMS

8.00000000E 02 1.64000000E 05 8.40000000E 06

## RANGE FOR VARIABLES

FXORIG

1.00000000E 02 1.00000000E 02 1.00000000E 02

FXLIM

1.00000000E 05 1.00000000E 05 1.00000000E 05

THERE ARE 6 EQUATIONS AND 6 UNKNOWN,CONSISTING OF 3 RESIST,  
THE LOWER BOUNDARIES FOR THE RESISTANCES, THE INDUCTANCES, AND  
AND 1.00000000E-11, RESPECTIVELY, WHILE THEIR UPPER BOUNDARI  
1.50000000E-01 RESPECTIVELY.

LX= 1

LX= 2

LX= 3

LX= 4

LX= 5

LX= 6

## VARIABLES

1.00000000E 02 1.00000000E 02 1.00000000E 04

SPECIFIED

ENRIES C( 1)= 7.93524923E-05 FARADS  
ENRIES

2

NA= 2

GRID=	1	NOS=	25
GRID=	2	NOS=	25
GRID=	3	NOS=	25
GRID=	4	NOS=	25
GRID=	5	NOS=	25
GRID=	6	NOS=	25
GRID=	7	NOS=	25
GRID=	8	NOS=	25
GRID=	9	NOS=	25
GRID=	10	NOS=	25
GRID=	11	NOS=	25
GRID=	12	NOS=	25
GRID=	13	NOS=	25
GRID=	14	NOS=	25
GRID=	15	NOS=	25
GRID=	16	NOS=	25
GRID=	17	NOS=	25
GRID=	18	NOS=	25
GRID=	19	NOS=	25
GRID=	20	NOS=	25
GRID=	21	NOS=	25
GRID=	22	NOS=	25
GRID=	23	NOS=	25
GRID=	24	NOS=	25
GRID=	25	NOS=	25

ALL ROOTS IN THE FOLLOWING SET LIE WITHIN THE PHYSICAL LIMITS

R( 1)=	2.53561570E	03	OHMS
R( 2)=	2.13625371E	03	OHMS
R( 3)=	5.61731033E	02	OHMS

L( 1)=	2.80865516E	01	HE
L( 2)=	2.84833827E	01	HE

LIMITS SPECIFIED

E 01 HENRIES      C( 1)\* 7.93518322E-05 FARADS  
E 01 HENRIES

2

COMMENCING COEFFICIENT APPROACH

NA= 1

GRID=	1	NOS=	25
GRID=	2	NOS=	25
GRID=	3	NOS=	25
GRID=	4	NOS=	25
GRID=	5	NOS=	25
GRID=	6	NOS=	25
GRID=	7	NOS=	25
GRID=	8	NOS=	25
GRID=	9	NOS=	25
GRID=	10	NOS=	25
GRID=	11	NOS=	25
GRID=	12	NOS=	25
GRID=	13	NOS=	25
GRID=	14	NOS=	25
GRID=	15	NOS=	25
GRID=	16	NOS=	25
GRID=	17	NOS=	25
GRID=	18	NOS=	25
GRID=	19	NOS=	25
GRID=	20	NOS=	25
GRID=	21	NOS=	25
GRID=	22	NOS=	25
GRID=	23	NOS=	25
GRID=	24	NOS=	25
GRID=	25	NOS=	25

ALL ROOTS IN THE FOLLOWING SET LIE WITHIN THE PHYSICAL

R( 1)=	2.53563727E 03	OHMS
R( 2)=	2.13621918E 03	OHMS
R( 3)=	5.61737470E 02	OHMS

L( 1)=	2.80868770
L( 2)=	2.84829220

3.40000000E 07      1.60000000E 04      1.20000000E 06

1.00000000E 01      1.00000000E 01      1.00000000E-02

1.00000000E 04      1.00000000E 04      1.00000000E-05

INCE(S), 2 INDUCTANCE(S), AND 1 CAPACITANCE(S).  
THE CAPACITANCES ARE 2.40000000E-01, 5.00000000E-05,  
S ARE 2.20000000E 07, 3.50000000E 02, AND

1.00000000E 01      1.00000000E 01      1.00000000E 03

2

NA= 3

GRID= 1 NOS= 25

GRID= 2 NOS= 25

GRID= 1 NOS= 48

GRID= 1 NOS= 96

USING THIS SET OF ESTIMATES, NO ROOTS WERE FOUND

NA= 4

GRID= 1 NOS= 25

GRID= 2 NOS= 25

GRID= 1 NOS= 48

GRID= 2 NOS= 48

GRID= 1 NOS= 94

USING THIS SET OF ESTIMATES, NO ROOTS WERE FOUND

NA= 5

GRID= 1 NOS= 25

GRID= 1 NOS= 50

USING THIS SET OF ESTIMATES, NO ROOTS WERE FOUND

COMMENCING CONSTANT APPROACH

GRID= 1 NOS= 25

GRID= 1 NOS= 50

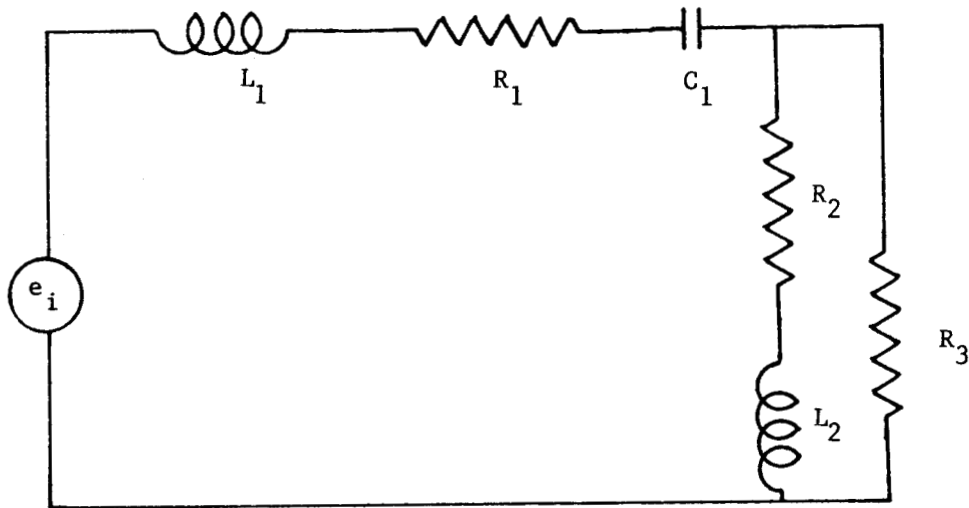
USING THIS SET OF ESTIMATES, NO ROOTS WERE FOUND



## APPENDIX D

### FILTER CIRCUIT WITH SIX UNKNOWNNS

#### D-1 Circuit Diagram



#### D-2 Identity of Unknowns

$$Y_1 = R_1$$

$$Y_4 = L_1$$

$$Y_6 = 1/C_1$$

$$Y_2 = R_2$$

$$Y_5 = L_2$$

$$Y_3 = R_3$$

#### D-3 Set of Equations

$$Y_4 Y_5 = 8.0 \times 10^2 \quad (D-1)$$

$$Y_2 Y_4 + Y_3 Y_4 + Y_1 Y_5 + Y_3 Y_5 = 1.64 \times 10^5 \quad (D-2)$$

$$Y_5 Y_6 + Y_1 Y_2 + Y_1 Y_3 + Y_2 Y_3 = 8.4 \times 10^6 \quad (D-3)$$

$$Y_3 Y_6 + Y_2 Y_6 = 3.4 \times 10^7 \quad (D-4)$$

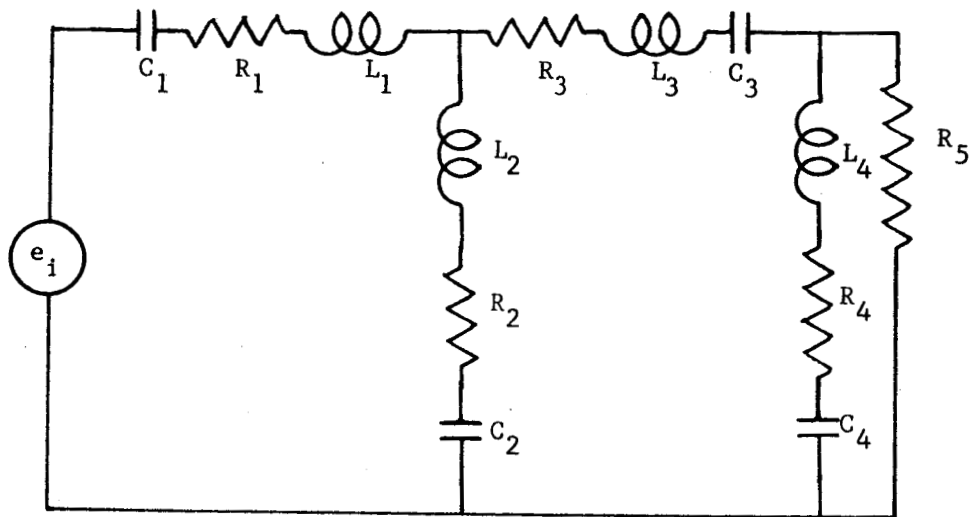
$$Y_3 Y_5 = 1.6 \times 10^4 \quad (D-5)$$

$$Y_2 Y_3 = 1.2 \times 10^6 \quad (D-6)$$

## APPENDIX E

### FILTER CIRCUIT WITH THIRTEEN UNKNOWNNS

#### E-1 Circuit Diagram



#### E-2 Identity of Unknowns

$$\begin{array}{lll} Y_1 = R_1 & Y_6 = L_1 & Y_{10} = 1/C_1 \\ Y_2 = R_2 & Y_7 = L_2 & Y_{11} = 1/C_2 \\ Y_3 = R_3 & Y_8 = L_3 & Y_{12} = 1/C_3 \\ Y_4 = R_4 & Y_9 = L_4 & Y_{13} = 1/C_4 \\ Y_5 = R_5 \end{array}$$

#### E-3 Set of Equations

$$Y_6 Y_7 Y_9 + Y_6 Y_8 Y_9 + Y_7 Y_8 Y_9 = 2.22 \times 10^5 \quad (E-1)$$

$$\begin{aligned}
& Y_4 Y_6 Y_7 + Y_4 Y_6 Y_8 + Y_4 Y_7 Y_8 + Y_5 Y_6 Y_7 + Y_5 Y_6 Y_8 + Y_5 Y_7 Y_8 + \\
& Y_2 Y_6 Y_9 + Y_3 Y_6 Y_9 + Y_5 Y_6 Y_9 + Y_1 Y_7 Y_9 + Y_3 Y_7 Y_9 + Y_5 Y_7 Y_9 + \\
& Y_1 Y_8 Y_9 + Y_4 Y_8 Y_9 = 5.965 \times 10^7
\end{aligned} \tag{E-2}$$

$$\begin{aligned}
& Y_6 Y_7 Y_{13} + Y_6 Y_8 Y_{13} + Y_7 Y_8 Y_{13} + Y_6 Y_9 Y_{12} + Y_7 Y_9 Y_{12} + \\
& X_6 X_9 Y_{11} + Y_8 Y_9 Y_{11} + Y_7 Y_9 Y_{10} + Y_8 Y_9 Y_{10} + Y_2 Y_4 Y_6 + \\
& Y_2 Y_5 Y_6 + Y_2 Y_4 Y_8 + Y_2 Y_5 Y_8 + Y_3 Y_4 Y_6 + Y_3 Y_5 Y_6 + \\
& Y_4 Y_5 Y_6 + Y_3 Y_4 Y_7 + Y_3 Y_5 Y_7 + Y_4 Y_5 Y_7 + Y_1 Y_4 Y_7 + \\
& Y_1 Y_5 Y_7 + Y_1 Y_4 Y_8 + Y_1 Y_5 Y_8 + Y_1 Y_2 Y_9 + Y_1 Y_3 Y_9 + \\
& Y_1 Y_5 Y_9 + Y_2 Y_3 Y_9 + Y_2 Y_5 Y_9 = 5.5399 \times 10^9
\end{aligned} \tag{E-3}$$

$$\begin{aligned}
& Y_2 Y_6 Y_{13} + Y_3 Y_6 Y_{13} + Y_5 Y_6 Y_{13} + Y_1 Y_7 Y_{13} + \\
& Y_3 Y_7 Y_{13} + Y_5 Y_7 Y_{13} + Y_1 Y_8 Y_{13} + Y_2 Y_8 Y_{13} + \\
& Y_4 Y_6 Y_{12} + Y_5 Y_6 Y_{12} + Y_4 Y_7 Y_{12} + Y_5 Y_7 Y_{12} + \\
& Y_1 Y_9 Y_{12} + Y_2 Y_9 Y_{12} + Y_4 Y_6 Y_{11} + Y_5 Y_6 Y_{11} + \\
& Y_4 Y_8 Y_{11} + Y_5 Y_8 Y_{11} + Y_1 Y_9 Y_{11} + Y_2 Y_9 Y_{11} + \\
& Y_5 Y_9 Y_{11} + Y_4 Y_7 Y_{10} + Y_5 Y_7 Y_{10} + Y_4 Y_8 Y_{10} + \\
& Y_5 Y_8 Y_{10} + Y_2 Y_9 Y_{10} + Y_3 Y_9 Y_{10} + Y_5 Y_9 Y_{10} + \\
& Y_1 Y_2 Y_4 + Y_1 Y_3 Y_4 + Y_2 Y_3 Y_4 + Y_1 Y_2 Y_5 + \\
& Y_1 Y_3 Y_5 + Y_2 Y_3 Y_5 = 1.77245 \times 10^{11}
\end{aligned} \tag{E-4}$$

$$\begin{aligned}
& Y_6 Y_{12} Y_{13} + Y_7 Y_{12} Y_{13} + Y_6 Y_{11} Y_{13} + Y_7 Y_{10} Y_{13} + \\
& Y_8 Y_{10} Y_{13} + Y_8 Y_{11} Y_{13} + Y_1 Y_2 Y_{13} + Y_1 Y_3 Y_{13} + \\
& Y_1 Y_5 Y_{13} + Y_1 Y_4 Y_{11} + Y_1 Y_5 Y_{11} + Y_1 Y_4 Y_{12} + \\
& Y_1 Y_5 Y_{12} + Y_2 Y_3 Y_{13} + Y_2 Y_5 Y_{13} + Y_2 Y_4 Y_{10} + \\
& Y_2 Y_5 Y_{10} + Y_2 Y_4 Y_{12} + Y_2 Y_5 Y_{12} + Y_3 Y_4 Y_{10} + \\
& Y_3 Y_5 Y_{10} + Y_3 Y_4 Y_{11} + Y_3 Y_5 Y_{11} = 1.81860 \times 10^{12}
\end{aligned} \tag{E-5}$$

$$\begin{aligned}
& Y_1 Y_{12} Y_{13} + Y_1 Y_{11} Y_{13} + Y_2 Y_{12} Y_{13} + Y_2 Y_{10} Y_{13} + \\
& Y_3 Y_{11} Y_{13} + Y_3 Y_{10} Y_{13} + Y_4 Y_{11} Y_{12} + Y_4 Y_{10} Y_{12} + \\
& Y_4 Y_{10} Y_{11} + Y_5 Y_{12} Y_{13} + Y_5 Y_{11} Y_{13} + Y_5 Y_{11} Y_{12} + \\
& Y_5 Y_{10} Y_{13} + Y_5 Y_{10} Y_{12} + Y_5 Y_{10} Y_{11} = 7.225 \times 10^{12}
\end{aligned} \tag{E-6}$$

$$Y_{11} Y_{12} Y_{13} + Y_{10} Y_{12} Y_{13} + Y_{10} Y_{11} Y_{13} = 9.0 \times 10^{12} \tag{E-7}$$

$$Y_5 Y_7 Y_9 = 9.0 \times 10^5 \tag{E-8}$$

$$Y_2 Y_5 Y_9 + Y_4 Y_5 Y_7 = 1.50 \times 10^8 \tag{E-9}$$

$$Y_5 Y_9 Y_{11} + Y_5 Y_7 Y_{13} + Y_2 Y_4 Y_5 = 6.78 \times 10^9 \tag{E-10}$$

$$Y_2 Y_5 Y_{13} + Y_4 Y_5 Y_{11} = 5.80 \times 10^{10} \tag{E-11}$$

$$Y_5 Y_{11} Y_{13} = 1.20 \times 10^{11} \tag{E-12}$$

$$Y_5 = 5.0 \times 10^2 \tag{E-13}$$

REFERENCES CITED

1. Kuo, Benjamin G., "Flow Graph Using Kirchoffs Law", Automatic Control Systems, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1962.
2. Mayeda, Wataru and Sishu, Sundaran, "Topological Formulas for Network Functions", Bulletin 446, Engineering Experiment Station, University of Illinois, 1956.
3. Freudenstein, Ferdinand and Roth, Bernard, "Numerical Solution of Systems of Nonlinear Equations", Association for Computing Machinery Journal, Vol. 10, March 1963.
4. Crandall, Dr. Stephen H., Engineering Analysis, McGraw-Hill Book Company, Inc., New York, 1956.
5. Buckingham, Dr. R. A., Numerical Methods, Sir Isaac Pitman & Sons, LTD., London, 1957.
6. Householder, A. S., Principals of Numerical Analysis, McGraw-Hill Book Company, Inc., New York, 1953.
7. Hartree, D. R., Numerical Analysis, Oxford Publishing Company, London, 1958.
8. Nielsen, K. L., Methods of Numerical Analysis, Macmillan Company, New York, 1964.
9. Saaty, T. L. and Bram, J., Nonlinear Mathematics, McGraw-Hill Book Company, Inc., New York, 1964.
10. Wilcox, J., Tung, S., and White, K., "Network Synthesis and a Method to Define a Transfer Function from Prescribed Phase-Magnitude Characteristics", NSL/Huntsville Aero/Astro Tech. Memo. No. 16, May 1964.
11. Ryan, Robert S., "Preliminary Stability Analysis of Saturn I Block II Vehicles", MTP-AERO-63-18, 7 March 1963.
12. Tatom, F. B., "Study for Simultaneous Solution of Sets of Nonlinear Algebraic Equations Occurring in Filter Circuit Analysis", NSL/Huntsville Technical Proposal NSL-HP-3B, December 1965.